

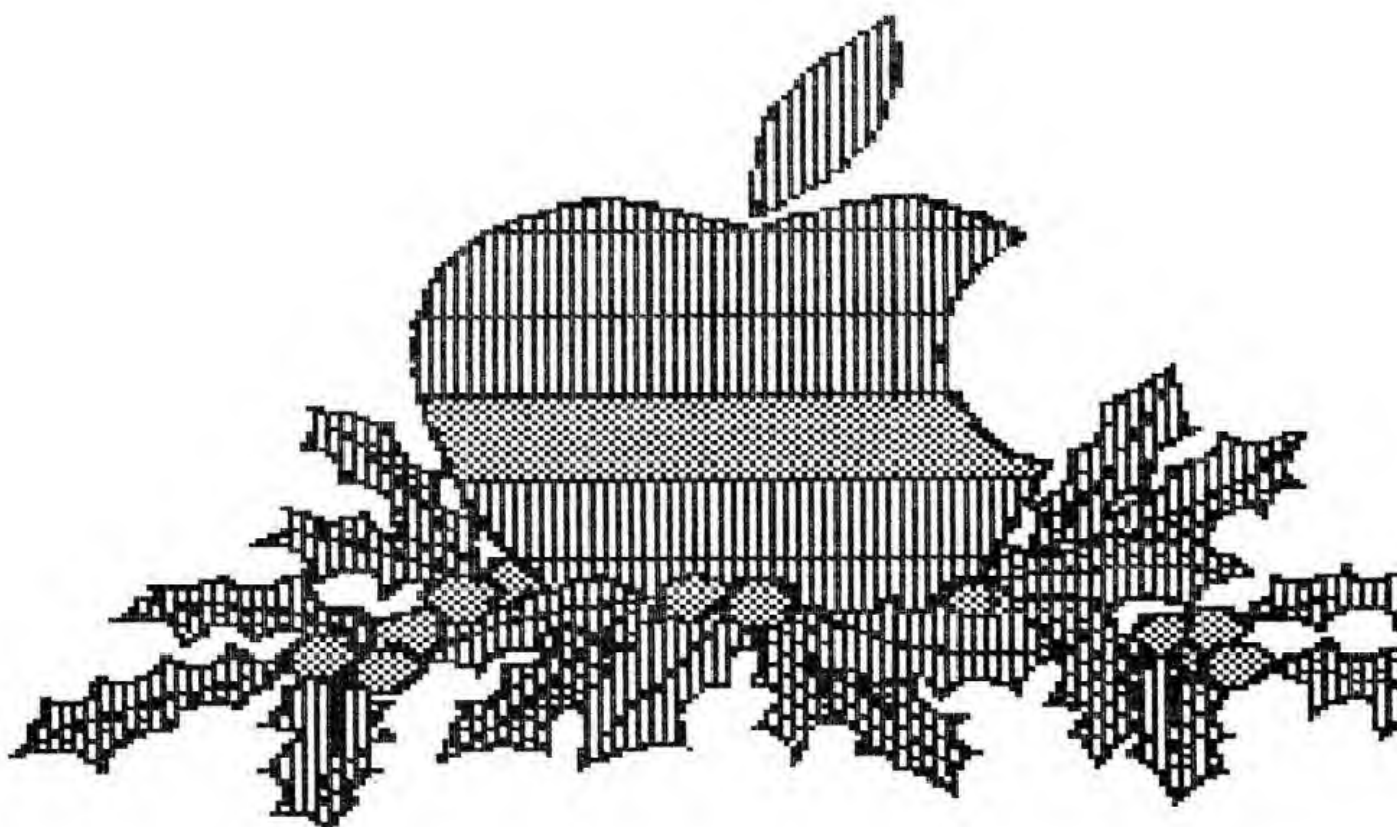
hard core

THE JOURNAL
OF THE
BRITISH APPLE
SYSTEMS
USER GROUP

DECEMBER 1981

£1

VOLUME 1 No 6



AN APPLED CHRISTMAS TO ALL
OUR READERS



For everything

apple .. ring**Lux Computer Services Ltd.**

108 THE PARADE HIGH STREET WATFORD WD1 2AW
 Telephone: Watford (0923) 29513

- Business Software • Professional Courses • Pascal •
- Recreational Software • Interfacing • Cromemco Systems •
- Level 1 Service Centre • Utilities • Colour Monitors •
- Supplies • Accessories • Enthusiasm • Open Tues - Sat •

**BLUE CHIP SOFTWARE**

10 Huson Close London NW3 3JW Tel: 01-722 8827

APPLE UTILITY PROGRAMS

by Ian Trackman

At long last - Apple have promised us that GO-BETWEEN (Standard Printer Version) will be available by the time that you read this advert !

With Go-Between, you can define your own printer commands and use them to control underlining, extended characters, etc. from anywhere within an Apple Writer file.

Go-Between also features a "screen print-out" to check your typing before committing it to paper, single- or twin-disk operation, lower-case compatibility, tabbing and much more !

£26.50

Still going strong :-

Go-Between (Centronics 737/739)

£26.50

Super Trace - a comprehensive Basic trace utility

£27.50

Super Editor - super-fast search and replace for Applesoft

£27.50

Packing Suite - packs, unpacks and strips Rems

£27.50

Symdis - a powerful symbolic disassembler

£22.50

hardcore

THE JOURNAL OF
THE BRITISH APPLE SYSTEMS USER GROUP
P.O. BOX 174 WATFORD WD2 6NF

EDITED BY DAVID BOLTON

BRITISH APPLE SYSTEMS USER GROUP

COMMITTEE

FRANK KAY	Chairman
JOHN SHARP	Secretary
DAVID BOLTON	Treasurer
JOHN RODGER	Librarian/ Information Officer
JOHN ROGERS	Software Librarian
TONY WILLIAMS	Membership Secretary

BANKERS

National Westminster Bank Ltd.,
Brompton Square, London SW3 1HH.

MEMBER OF THE
INTERNATIONAL APPLE CORE

COPYRIGHT (C) - The contents of this journal are copyright of the British Apple Systems User Group and / or the respective authors. However permission will normally be granted for non-commercial reproduction by user groups affiliated to the International Apple Core, provided the author and source are properly credited.

The opinions and views expressed are those of the various contributors, and are not necessarily supported by the British Apple Systems User Group.

CONTENTS

4 Editorial	
5 The Software Library	John Rogers
6 Beginner's Pages	John Sharp
8 Courses	
8 The First Noel	Ann Padley
11 BASUG at the Dutch Computer Convention	Tony Williams
14 Diskourse	Frank Kay
16 Applesoft 'Input Anything' Updated	Ian Trackman
21 DOS Command Syntax Table	Cliff Wooton
24 Pascal Pages	Warren Avery
27 PCW Fair	Tony Williams
28 ITT Ninth Bit Addressing	Ken Gordon
31 Pippin's Page	Vernon Quaintance
34 Applesoft Compilers Compared	reprinted from Creative Computing
41 Readers' Letters	
45 Book Reviews	
46 Apple Panic	reviewed by Vernon Quaintance

ADVERTIZERS

2 Lux Computer Services Ltd
2 Blue Chip Software
5 Wida Software
7 Microsource (Jailbreak)
8 Lion Microcomputers
9 The Byte Shop/Computerland
10 Spider Software
15 K.J.Studios
18 Personal Computers Ltd
19 S.B.D. Software
19 Robocom Ltd
20 B.F.I. (Verbatim)
23 Systemics Ltd
26 Autocrat Computers
28 Portatel Conversions Ltd
29 Accutrak Disks
30 C/M Computers
32 S.B.D. Software (Call-APPLE)
33 Microsource (Business Basic)
33 K.J.Studios (Plugit)
38 Apple Fan Special Offer
39 Hi-Teck (Apple Fan)
40 Mass Micros
44 Brucetel (Tantel)
45 Microsource (What's Where)
46 Microsource (Type-Right)
47 Pete & Pam Computers
48 Micro-Peripherals Ltd

Cover Picture - Another pictorial masterpiece from the versawriter of Norah Arnold.

EDITORIAL

With this issue we come to the end of the first volume of 'Hard Core'. The production of the magazine has not been without its problems - indeed each issue seems to have been accompanied by its own crisis - but from the response we get it appears that Hard Core is meeting a need. Without this encouraging response the magazine could well have foundered before now.

The positive response is not only from readers - you'll notice in this issue contributions from several new authors, as well as more from the stalwarts. But don't (please!) take this to mean that you needn't bother - it takes a lot of material to fill a magazine every two months.

The actual production work continues to be a problem, which is considerably eased if we have plenty of articles in the bank (if we had a couple of month's buffer of articles we could share out the editing/proofreading burden more widely).

Another high spot is that that advertising is picking up. Because the magazine is produced entirely on a voluntary basis, we don't do much chasing of advertisements, and potential advertizers, in fact have to work quite hard to get an advert into Hard Core. That they are willing to do this says a lot about the reputation of the magazine and indeed the group.

Despite the problems and the intense time pressures I've greatly enjoyed editing Hard Core and generally helping to get the group established. I'd like to thank all the other people who've helped and ask that this help continues hopefully from an ever-increasing group of people in 1982.

In this connection (!) would all of you who've got access to a Prestel terminal let us know, as we're investigating the possibility of distributing software and perhaps even work (see the comment above on sharing the magazine workload) by means of a telephone link of some kind. Please state whether you have a link between your Apple and the Prestel terminal, and also let us know if you have a modem.

IN THIS ISSUE

You'll notice that for the first time we've re-printed an article from an American magazine. This is something that we thought we should find room for - Basic compilers are a new development, they fill a definite need, four are now on the market, all substantially different - and a large proportion of recent queries are about these beasts! Hence this article, which as you will see is unfortunately already out of date.

The other thing to say about this issue is that I didn't edit it! It has been jointly edited by Tony Williams, John Sharp and myself, with substantial help from John Molloy, Fran Teo, Jane Davies and Stuart Morley.

WHAT'S HAPPENING

The last update gave advance news of "Apple 82" - the weekend convention to be held in April 1982. We are sorry to report that this particular Apple has gone sour.

This event was originally conceived as a user convention and Apple UK (Microsense as it then was) approached BASUG and Windfall with a request that we jointly organise this with Apple providing the funding.

Windfall wanted to organise on their own behalf a money-making exhibition to be held in conjunction with this convention and although Apple originally agreed to this, the commercial trend of the affair together with policy changes consequent upon the take-over of Microsense by Apple led them to withdraw their sponsorship. In these circumstances it was decided that BASUG would not participate in this event but would instead revert to our previous idea of holding a weekend convention later in the spring in conjunction with our AGM.

Welcome to our 901st member! And guess who? Apple Computer Ltd, of Cork City, Ireland. Do you know something that Apple UK doesn't?

PRODUCT NEWS

Lion Microcomputers are at present offering a very good deal on ITT 2020 computers which come together with a disk drive for £699 + VAT. As long as you appreciate the differences between the machine and an Apple this could be an extremely appealing proposition for school use, as a dedicated business system or as a back-up system.

The PEAR II (only it's not called that any more) has finally been launched and was on show at COMPEC. Basically it's an Apple, with 14 slots, full Upper/Lower case keyboard, but without (for copyright reasons) any firmware. Still this will only knock you back £40 or so, making it (assuming it is as good as it looks and we haven't tested it yet) an attractive alternative at a trifle over £1000 to a standard Apple at £812. Look out for a new DOS soon (DOS 4.0 perhaps). We'd expect this to support larger disc storage (8" and Winchester) as well as expanded memory options.

It appears that early in '82 Apple planned to make a European keyboard with Upper/lower but were told there was no demand for it. Whoever did they ask?

It seems that with D C Hayes showing a positive lack of interest in Europe, the Apple-approved modem for the European markets will be the Cat Novation. Typically, the French telephone authorities approval of this device hinges on it being made in France, and this it seems will happen.

Pascal Utilities are, Apple say, coming soon.

David Bolton

THE SOFTWARE LIBRARY

By John Rogers

Firstly we must apologise for the delays in fulfilling software orders. Normally it would take about 2 - 3 weeks from posting your order to receiving the disks. This does depend on the timing of the various stages e.g. order posted on a Friday and/or not enough time to process it immediately. Recently the time taken to process orders has increased due to two main factors. One, the flood of orders received for disks 20 - 23 and 30 - 34 and, secondly, co-ordination problems within the process.

With the above situation and the prospect of the Christmas post looming up, orders will be taking longer to process. Again we are sorry for this, but we are doing our best in the time available to us to carry out this voluntary work.

The use being made of the C.S.L has very much decreased recently, one of the main reasons being the inclusion of programs from the C.S.L into the S.D.L. It has therefore been decided to withdraw the C.S.L for obtaining programs. Please note that it WILL continue to receive contributed programs from members on the same basis laid down, ie. the credit system will continue in the same way. Programs currently in the C.S.L, and all future submitted programs, will be formed into whole disks and then form part of the S.D.L.

We are currently in the throws of producing a booklet (or booklets) that will give short descriptions of the programs in the software library. There are one or two people who have volunteered to help compile such a booklet, but as can be seen from the growing library, a lot of work is required. Therefore we would like people will to commit their thoughts and observations on programs to Apple-Writer files. If you are willing to help, please drop me a line then we can discuss more fully what is required.

As a final note, we wish to clarify the point that only members who have contributed software to the library and are in credit, are eligible to the discount or credit as in the August edition of Hardcore.

Wida Software

Specialists in Educational Software for Schools

APFELDEUTSCH.....£ 120.00
CAL course in German; Beginners to O'Level and beyond. Textbook: Workbook: 6 Language Lab cassettes: 9 Apple disk of teaching and testing routines.
Apple only. (20% discount for schools)

TEACHER'S TOOLKIT.....(Apple disk) £ 20.00
(Pet cassette) £ 15.00
Starter pack for building up library of tests. No knowledge of computing needed. Suite of five programmes; Tester, Testmaker, Editor, Multiple Choice specimen, Directory of tests

LERARENPAKKET.....(Apple disk) £ 25.00
Teacher's Toolkit as above but with on-screen instructions in Dutch

GERMAN ROUTINES.....(Apple disk) £ 15.00
(Pet cassette) £ 10.00
Stand alone teaching and testing routines - article and adjective endings, word order, pronouns, etc. Send s.a.e for details

PEDAGOG AIDS.....(10 Apple disks) £ 60.00
Life raft for teacher thrown at the deep end of Computer Studies. 20 Apple programs taking you to O'Level.

SHAPE MANAGER.....(Apple kit) £ 59.65
The easy way to build up shapes and use them in your program. Does for shapes what a word processor does for words. From Sinta Software

TYPE-RIGHT.....(Apple kit) £ 46.50
Lower case word processing with Apple Writer. Full shift key operation. Plug-in fitting with disable switch. Manual, fitting instructions, software on disk.

APPLE PILOT.....(2 disks) £76.00
The ultimate language for teachers; mix sound, graphics and text on-screen in your teaching programmes. And much else. Disks and manuals

ARISTOTLE'S APPLE.....(Apple disk) £ 20.00
Authoring system; easy to use multiple choice matching tests. Includes alternative answers.

LUDINSKI'S LOGS.....(Apple disk) £ 20.00
O'Level revision of logs problems. Teaching, testing and scoring. Correct solution rewarded by 25 seconds of arcade game.

FRENCH AND GERMAN CHIP(for Pet) £ 35.00
Plug-in replacement chip for (New ROM) Pet Gives Umlauts, accents etc. Easy to fit and use. Full instructions

THE LOWBROOK TAPES.....(Pet cassette) £ 10.00
Suite of six numeracy programmes for the primary school (fractions, carrying over, etc.)

COMPUTERS FOR KIDS.....(book) £ 4.00
A Basic programming manual written just for kids. Special section for teachers and parents. by Sally Greenwood Larsen Apple II plus edition

All prices inclusive of VAT postage & packing free

Contact

Wida Software
The Red House
2 Nicholas Gardens
London W5 5HY
Tel (01) 567 6941
or (062 882) 5206



software

BEGINNER'S PAGES

by John Sharp

A number of beginners have been writing in requesting help with binary files, so this month let's begin with a look at what files you might possibly have on a disk.

If you catalog a disk you will get something like this:-

```
A 002 HELLO
I 020 DRAGON MAZE
A * 004 PROGRAM 1
B * 002 FILE 1
T 003 TEXT FILE
R * 005 ANOTHER FILE
```

You may not get all these on one disk, and indeed you may not even have seen a T or R or even B on your disks. Leaving aside the first column of A,I,R,T,B for the moment, what does the rest mean? Well: the * tells you if a file is locked. If it is locked, you cannot write to the disk using the same program name. That is, you cannot wipe out your program on the disk by overwriting it. The number denotes the number of sectors the program takes up on the disk. As a rough guide each sector is 0.25K. The name to the right is just that, the name of the file. I say file because it is that rather than a program. It is only when entered correctly into the Apple that it becomes a program, if that indeed is what it is. A file with a T in the first column, is not a program. It might be a TEXT file to EXEC or it might be, as the prefix suggests, a TEXT file. A TEXT FILE is a bunch of data ready to be read into a program, e.g. a set of records, such as a set of names and addresses. If you want to see what is in a TEXT file then type:-

```
MON C,I,O    <RETURN>
EXEC XXXXXXX <RETURN>
```

where XXXXXXX is the name of the TEXT FILE you want to look at. As each bit of data is printed, since you are doing the equivalent of typing in directly from the keyboard and pressing return, you will get a SYNTAX ERROR, just as you would if you typed a command the APPLE does not understand. You can slow down the screen printing by using CTRL-S. It tends to be a little noisy because of all the beeps and syntax errors, but can be a very useful tool. Alternatively, you could use the READ TEXT program on the master disk.

The A and I program file names are fairly straightforward. They correspond to APPLESOFT

and INTEGER program files respectively. Unless you are only running disks that boot and take over the machine, you will be familiar with both of them.

The problem arises, however, with B (and R) files. B stands for BINARY FILES. R stands for RELOCATABLE FILES which are a special type of Binary file. You will see some on the DOS 3.3 TOOLKIT disk. They are produced from the APPLE ASSEMBLER on this same disk. So with one slight difference, they are basically the same type of file, a saving of a batch of machine code on disk. This can have various functions once it is in memory and this causes the problems for beginners.

The first type of file it can be is a program, written in machine code to make it run faster. A program example is a FID, MUFFIN or MASTER CREATE (on DOS 3.3 MASTER), and UPDATE 3.2.1 (on the DOS 3.2 MASTER). In order to get these to run as programs, simply BRUN FID or whatever the program name is.

Before going further, let me say a word about the number 034 or whatever is the number of sectors the file takes up on the disk. It only helps you (and the DOS) to keep track of how much of the disk you are using. This is dealt with in the DOS manual so I will not go into it here. For those members with tape only it will help to explain what these numbers mean in the software library lists. As a rough guide, four sectors equals 1K of program.

The next type of file is a Hi-Res Picture that has been saved. The length of memory taken up by a Hi-Res Picture is 34 sectors; so if you see :-

```
B * 034 BASUG LOGO
```

on a disk catalog, you can be fairly confident that it is a picture. There may be other binary files coincidentally 34 sectors long which are not pictures, but the name usually tells you they are not. Alternatively, there are now ways of compressing pictures, so a picture can be less than 34 sectors. You then need a special program to put them back on the Hi-Res pages correctly. If you try to BRUN a Hi-Res picture, anything could happen. Normally you will just halt in Monitor. Try a few and see. You will not do the Apple any harm, just confuse it.

The third type of binary file is a set of data used in a program. It is a little bit like a text file except that it is totally machine code and would mean nothing except to the program that uses it. The best example is a shape table, which is a set of points and directions. If looked at other than a shape table by the program you are using, it is

totally meaningless. A good programmer will put some indication such as MARTIANS.OBJ or MARTIAN.SHAPE to let you know it is used in another program. The DOS 3.3 TOOLKIT fonts for example are labelled BYTE.SET, ROMAN.SET etc, to make this clear. It is as important as putting REM statements in if you wish to let others know what is going on. On some of the software library disks there are binary programs with just a single letter or a pair of letters; these are fairly obviously used by another program, and the author has made them too short to make you think to run them; the name just doesn't mean anything.

Another type of set of machine code might be data, as for example in the COPY program on the 3.3 Master disk. The copy programs look like this:-

* I 009 COPY

* B 003 COPY.OBJ0

* A 009 COPYA

There are versions for you to use for copying disks if the BASIC you are using is INTEGER (the first one) or APPLESOFT (the last one, which has an A tagged on the end since two programs cannot have the same name on the same disk otherwise there would be confusion when it came to running them.) In between is a machine code set of data for the other programs (both of them) to use. If you list these programs you will see a line that has a print "BLOAD COPY.OBJ0" at line ... Why no D\$=CHR\$(4), well there is an invisible CTRL-D. You could see this if you used something like The APPLESOFT PROGRAMMERS ASSISTANT on the DOS 3.3 TOOLKIT, or the PROGRAM LINE EDITOR.

The most confusing programs on the 3.3 Master it would seem are catalogued as:-

B * 050 FPBASIC

B * 050 INTBASIC

They are in fact APPLESOFT (or Floating Point BASIC) and INTEGER BASIC, respectively. If you have a language or RAM card, these files will be loaded onto the language card and the card locked so that it appears to be an INTEGER CARD or APPLESOFT ROM CARD. If you have an APPLE II PLUS, look at the HELLO program, by just loading it. LINE 210 has "BLOAD INTBASIC,A\$D000". Again there is an invisible CTRL-D. The A\$D000 means load it at position D000 in memory, which is on the language card.

When a binary file is saved, it is necessary to tell DOS the start and ending locations of the program. This information is saved onto the disk. When you load the program back, it will be loaded into the same position, unless you tell it otherwise. The A\$D000 in this BLOAD statement

tells it not to load it where it was saved from. This is in fact in the middle of memory and it will load there if you just BLOAD INTBASIC. If you try to BRUN INTBASIC, since it is not written to run in this location, but on the language card, then it will cause you to think your machine has developed a fault.

If there are any more problems, please write in and we will see what help we can give.

THE LAST WORD

Finally, for those of you who are not members of the British Apple Systems User Group, we welcome all users of Apple Systems (Apple, ITT, and Pearcom machines). Membership for the calendar year 1981 (including all back issues of the magazine) costs £10, and 1982 renewal subscription is also £10. If you join for the first time in 1982, there is a one-time application fee of £2.50. For your subscription you get an introductory disc, subscription to Hardcore, access to software and book libraries, and regular meetings in various parts of the country. Simply send your cheque to BASUG, P.O.Box 174, Watford, WD2 6NF.

JAILBREAK ITT

COME OUT FROM BEHIND THOSE HI-RES BARS

Now you can run any hi-res program written for your Apple on your 2020 without the annoyance of the 40 vertical bars. Chess and complex games are a joy to behold.

* Saves having to modify programs to 2020 standard (no mean task with complex games which can be over 16K bytes long).

* No soldering required - Fit in five minutes by following the simple instructions.

* Switchable between Apple and 2020 modes.

* Compatible with Apple PASCAL and Microsoft Softcard BASIC.

* Gives 280X192 hi-res dot screen like an Apple (the 2020 normally gives a 360X192 hi-res dot screen). Text and lo-res are unaffected.

* Does not support hi-res colour. Colours produce various shades of grey on a b&w display.

£33

inclusive.

MICROSOURCE

1, BRANCH ROAD, PARK STREET, ST.ALBANS.

TEL: 0727 72917 & 09273 75093

COURSES-COURSES

We apologise for the delay in replying to your letters regarding the courses outlined in the last edition. Unfortunately, because he has started a new job, Eddie Payne has had to resign from the committee. Many thanks for all the work you put in Eddie, especially at the PCW show, and best wishes for your new job, life, incarnation? The fact that Eddie has come to his senses has put a strain on the other members at a time when there has not been much free time and many pressing items to be dealt with. At the moment, we are hoping to run the BASIC for Beginners and Machine Code courses, with details sorted out (we hope in UPDATE 6) and the PASCAL COURSE of course! The PASCAL course, as postponed from early this year, will be run in June. It will probably be a three day-long weekend, and accommodation will be available. The Business software could be run if someone is prepared to lead/organise it. We would also like someone to lead a workshop/course for hardware, e.g making your own cards. Any volunteers for either of these?

THE FIRST NOEL

by Ann Padley

This should be a very useful program to have standing by at this time of year. It will almost certainly drive your in-laws out of the house and will probably annoy carol singers greatly, as it goes much faster than they do. By halving all the values in data statements 400,401 and 402 you can make it go very fast indeed. By selecting random 2-figure numbers for data statements 200, 201 and 202 you can make strange noises. Best of all, by adding a 'GOTO 500' statement at the end you can have continuous background music all through Christmas Day.

```

10 POKE 768,173: POKE 769,48: POKE
   770,192: POKE 771,136: POKE
   772,208: POKE 773,5: POKE 77
   4,206: POKE 775,81: POKE 776
   ,3:
20 POKE 777,240: POKE 778,9: POKE
   779,202: POKE 780,208: POKE
   781,245: POKE 782,174: POKE
   783,80: POKE 784,3: POKE 785
   ,76:
30 POKE 786,0: POKE 787,3: POKE
   788,96:
50 DIM T(72)
60 DIM D(72)
100 FOR L = 1 TO 72: READ T(L): NEXT
    L
200 DATA 101,112,128,112,101,95
   ,83,75,67,64,67,75,83,75,67,
   64,67,75,83,75,67,64,83,95,1
   01
201 DATA 101,112,128,112,101,95
   ,83,75,67,64,67,75,83,75,67,
   64,67,75,83,75,67,64,83,95,1
   01
202 DATA 101,112,128,112,101,95
   ,83,64,67,75,75,83,64,67,75,
   83,75,67,64,83,95,101
300 FOR K = 1 TO 72: READ D(K): NEXT
    K
400 DATA 50,50,150,50,50,50,200
   ,50,50,100,100,100,200,50,50
   ,100,100,100,100,100,100,100
   ,100,100,200
401 DATA 50,50,150,50,50,50,200
   ,50,50,100,100,100,200,50,50
   ,100,100,100,100,100,100,100
   ,100,100,200
402 DATA 50,50,150,50,50,50,200
   ,50,50,200,100,250,100,100,1
   00,100,100,100,100,100,100,2
   00
500 FOR L = 1 TO 72
600 POKE 848,T(L): POKE 849,D(L)
   : CALL 768
700 NEXT L

```

**Best Value Word Processing
In Town!****"A BARGAIN"****Financial Times 16.10.81**

Tick Box

☐**£699**

* Powerful ITT 2020 Micro
Computer with disc drive
and word processing
software

☐**£899**

* As above, with silent printer

☐**£1,429**

As above, with letter quality
printer and green display
screen

Prices
Exclude V.A.T.

* Works with any T.V. or
Home Monitor

FULL CREDIT TERMS AVAILABLE

Training and Maintenance all available ring now on our 24
HOUR HOT LINE - 01 636 9613 to find out more

Or cut out the ad, fill in
the details and send to:-

Name _____

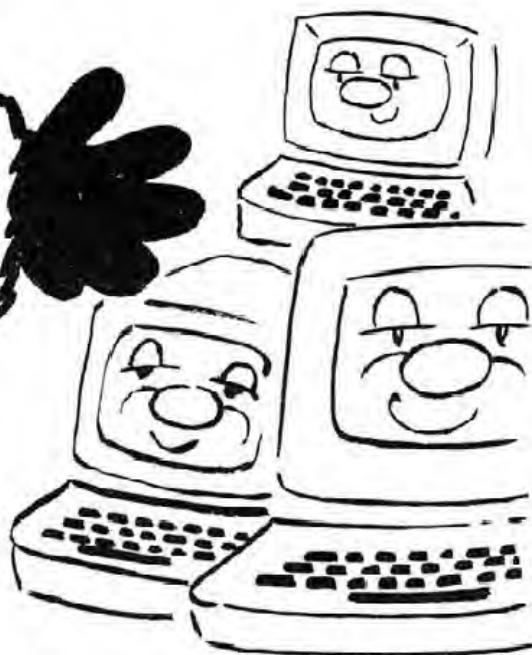
Address _____

Lion Micro Computers
227 Tottenham Court Road
London W1P 0HX

SEASONAL GREETINGS



FROM



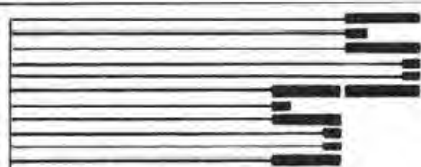
THE BYTE SHOP COMPUTERLAND

Special offer for B.A.S.U.G. members

BMC 12" monitor	£119.50
Green phosphor, 18MHZ bandwidth	
Wabash disks (box 10)	£15.00

Great deals for the month of December on ALL Apple listed products and various printers.

324 Euston Rd, London NW1 3BG. TEL 01-387-0505
 PRICES EXCLUDE VAT. VISA + ACCESS ACCEPTED



Spider Software

Alien Rain (Galaxian) - Colour + hi-res version of pub game **£13.95** M48K

Alien Typhoon - A much more difficult version of Alien Rain **£13.95** M48K

Apple Panic - Chase little apples up and down ladders **£16.95** M48K

Autobahn - Road race game with sound and hi-res graphics **£14.95** M48K

Both Barrels - 2 hi-res action games on one disc **£12.95** M48K

Cranston Manor - A new hi-res adventure like Wizard & the Princess **£19.95** M48K

Cyber Strike - 3-D hi-res action adventure in space **£19.95** M48K

Demon Derby (Hyper Head-On) - 4 skill level hi-res car race game **£13.95** M32K

EXPEDITER II - THE APPLESOFT COMPILER AT A LOW, LOW PRICE **£56.95** A48K

Galaxy Wars - Colour graphics + Sound effects + Hi-Res **£13.95** M32K

Gamma Goblins - Yet another superb hi-res action game **£16.95** M48K

Gobbler - Eat up the dots but watch out for the Gobblers **£13.95** M48K

Gorgon - Another superb pub game for your Apple **£21.95** M48K

Hi-Res Cribbage - The titled describes it. Even hear the pegs move **£14.95** M48K

Hi-Res Soccer - English football for 1 or 2 players in hi-res **£16.95** M48K

Missile Defence - Hi-res animation and sound arcade game **£16.95** M48K

Mission: Asteriod - Hi-res adventure in 21 colours. Save the World! **£10.95** M32K

Mystery House - Hi-res adventure using over 100 pictures **£12.95** M48K

Oldorf's Revenge - Hi-res adventure with 100 rooms. **£11.95** A48K

Operation Apocalypse - Four computer simulations of World War II **£35.95** A48K

Orbitron - Fight off enemy forces and avoid meteor showers **£14.95** M48K

Paddle Graphics - Draw your own pictures in 21 hi-res colours **£21.95** A48K

Pool 1.5 - Hi-res colour graphics pool table simulation. 4 games **£19.95** M48K

Pulsar II - 2 superb hi-res games on one disc **£14.95** M48K

Sneakers - Waves of little creatures attack you in hi-res **£16.95** M48K

Snoggle (Puckman) - Hi-res maze of ghosts. Great fun **£13.95** M48K

Space Eggs - Hi-res super-fast arcade style game **£14.95** M48K

Space Warrior - Hi-res pub game **£13.95** M48K

SUPERSCRIBE - WORD PROCESSOR. TRUE UPPER/LOWER CASE ON SCREEN **£73.95** M48K

Tarturian (Wizard) - Another hi-res adventure with 160 rooms **£14.95** A48K

The Shattered Alliance - Swords & Sorcery on a far-flung world **£35.95** A48K

The Warp Factor - 1 or 2 player hi-res Star Trek/Space War game **£23.95** A48K

The Wizard and the Princess - Hi-res adventure in 21 colours **£18.95** M48K

A: Requires Applesoft in ROM.
M: Will run on any Apple II.

WE DON'T JUST SELL GAMES

We can also supply Visicalc, Wordstar, Z80 Softcard, 16K Ramcards, etc at very competitive prices. If you want anything Apple please contact us.

WE ALSO OFFER A BESPOKE SOFTWARE SERVICE WHICH IS SECOND-TO-NONE

Prices include VAT at 15%. Add 50p p&p for orders under £30 totally.

Please write or telephone for your free copy of our up-to-date software list.

DEALER INQUIRIES INVITED. PERSONAL CALLERS BY APPOINTMENT ONLY PLEASE

Spider Software

98, AVONDALE ROAD, SOUTH CROYDON, SURREY.

Telephone: 01-680 0267 (24 hours a day — 7 days a week)

BASUG AT DUTCH COMPUTER CONVENTION

by Tony Williams

The weekend of November 27-29 saw a mass migration of BASUG members through Belgium (lemmings through the flemings!) toward Utrecht where the 6000-strong Dutch Computer Club was holding its annual get-together in the Juliana Hall.

This is my hastily written report.

Let me first give the club's thanks to the Dutch Club for their hospitality and for their great organizational skill (i.e., you didn't notice it). They have hit upon the very, very successful formula of taking over a great barn of a hall, erecting sturdy wooden market stalls (covered, against the rain?) and renting out space very generously at a low, low price. Just as important, the broad thoroughfares between rows of stalls gave the ten thousand or so visitors ample room to spread their wings. The English shows are mean, over-inflated and money-grubbing by comparison. The catering ranged from posh restaurant to inexpensive but good sausage stalls. BASUG would be proud to mount a show of this calibre.

What were you all doing there and how could BASUG afford it? How come you people down there in the southeast are off spending members' money on a foreign junket? you will be asking. Answer! we didn't. Our twelve person delegation went at our own expense, fares, petrol, hotel bills etc. However, because we were carrying a full carload of BASUG materials, including six cartons of Hard Cores, for instance, we felt justified in paying the hovercraft fare and petrol of one of the three vehicles. Point two: it wasn't a junket - a hard day's drive there, a full day on the BASUG stand, a day driving back - not really a fun trip. Results! thirty odd members recruited, all the magazines sold, BASUG's name made known. Write in and let us know if you disapprove of this kind of enterprise - and we will send the boys round to knock on your door.

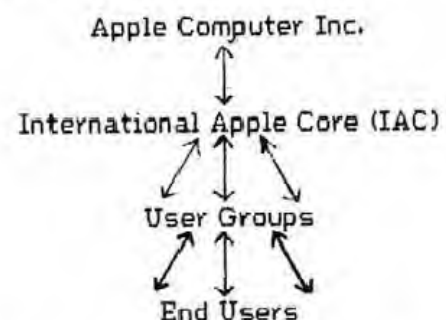
More important, however, was the personal contact made with members of European Apple groups big and small. This brings us to another pressing reason for our presence in Utrecht.

The only off-exhibition event to take place was the conference of the European IAC in the Beatrix Hall. This was organized by the European president of the IAC Wolfgang Dederichs, with the intention of bringing the

European Apple Users groups together for the first time and jointly confronting representatives of Apple Paris. This was largely successful in the first part, less so in the second. Apple wisely chose to fund this one event, but unwisely considered it not important enough to send along their top men. Perhaps this was because they think very little of user groups in general, but also because they are still failing to take account of the real needs of the public - and the measurable benefit of user groups to Apple itself.

Let's go back to the beginning. Wolfgang Dederichs frankly admits that he got the job of European director because he was the only European to attend the Chicago International Apple Core convention last year. 'International' you will remember, means outside California. "You're from Europe, eh? Okay, you be Director, Europe". Fair enough. But director of what? that is the question. There is no IAC Europe organization, and as Wolfgang explained, in any case the IAC is presently embroiled in a legal wrangle with Dilithium Press (which publishes Apple Orchard which finances the IAC). This has effectively meant that the IAC's funds have been cut off and is apparently the reason why we can't get regular supplies of Apple Orchard.

So if IAC is to get off the ground in Europe we have got to do it ourselves, if we want it to. In his opening remarks Wolfgang gave us an historical overview of the "top-down" structure we are all part of:



Historically, we are told, the IAC set up or somehow gave rise to the User Groups and now information is disseminated downwards and upwards via the IAC. All a load of codswallop, of course. As many of us know, BASUG constituted itself with no outside inspiration or help and we merely paid our dues to join the IAC's Faraway Tree. Information we scabble together ourselves, and whenever a meeting of importance has been arranged we have

invariably been informed too late to participate, if at all. What goes for BASUG goes for all the European groups represented at the meeting. So we have a rather jaundiced view of future prospects. In brief, our attitude is to maintain a positive if cautious watching brief: we think cooperation is a good idea, and are in favour of the IAC too, if properly organized. We will respond 'from below' to any initiatives from Apple 'above'.

As our very own Richard Zawadski so forcefully put it, at the exhibitions and fairs throughout Europe the user groups are standing in the front line for Apple. Prospective buyers are checking with us first before making their decision. But, as G. van de Woude also pointed out, Apple responds to this in the most cynical way - "Great, carry on the good work fellas!" (Aside: "Suckers! We just love an unpaid volunteer sales force!"). Not good husbandry. Lack of appropriate recognition could turn us sour on Apple and this could make a perceptible dent in Apple's sales profile. After all, as C. Bongers reminded us, the Pearcom has shown itself to be very responsive to European requirements and it has produced what is in many respects the Apple III we actually wanted instead of what we got. (I had thought that the whole purpose of the trip was to bring a Dutch manufactured Pear for the club, but I only succeeded in getting my wrist slapped).

Catching the flak all this time was Howard Stegner from Apple Paris. He did a remarkably good job for them, all things considered, although he was really given a going over. Let me explain. Howard is on the technical side and was sent to tell us about new products. There is only one new product, the Apple III. However Howard was able to give a very competent and detailed description of what it does and what went wrong the first time around. But he was trying to "convince the unconvincible" (R. Teed). The point is that whereas Apple II was aimed at the home user, it was quickly taken up by the small businessman and educationalist, etc., with the result that the home user now forms only a thin wedge of the Apple market. Reflecting this, Apple III has deliberately been designed and priced to cut the home user and hobbyist out. Although overall compatibility with Apple II was aimed at, woe betide any user with Mountain Hardware boards, or the Sup'R Terminal - their cards are simply too long!

Where does it all leave us? Out in the cold! Plans to bring out European editions of the

Apple II with easily accessible lower case and special characters for foreign word processing have been officially shelved. Apparently it is felt that the ingenuity of outside manufacturers is enough to make up for the Apple II's inbuilt drawbacks, its lack of true shift upper and lowercase operations, etc. The fact that these are all stopgap fiddly adaptations is a fact of life that has to be lived with. The Apple III, it must be said, does have lower case access and user definable foreign character sets, etc, which just about brings it into line with every low cost micro to be seen nowadays.

To summarise, the mood of the meeting was one of pessimism vis-a-vis the prevailing corporate attitude at Apple and a feeling that the company is less than adequately equipped in products or ideas to face the developments of the next two or three years. If Apple is to count on the support of the user groups it will have to reveal a hitherto concealed 'user-friendly' face.

Space and time forbid a more detailed review of all the other talks given during the day.

Much more positive things emerged from the conference. We were able to get a first taste of the way other Apple user groups do things.

We already know that AUGÉ (Apple User Group Europe - they really have to do something about that name; they are the German-speaking user group and should not claim to be otherwise,) has a pretty refined decentralised structure - which made it possible for them to send along members of their local regional branch (a two-hour drive). At BASUG we haven't been able to get around to tackling this problem systematically yet, but after all, they are two years older than us. AUGÉ has some 1100 members.

One of the most promising areas in which BASUG can be of some use is in assisting the fledgling 30-strong Club Micro Informatique Apple from Paris. This club has taken a completely different tack from ours: their members pay a very high membership fee (800Fr, I think, for which they receive lessons in BASIC - making it much cheaper than commercial courses). Virtually all members are brand new to Apple and need a lot of instruction and self-help. This is an idea worth emulating. This tiny group represented by Judith Kertesz (responsible for "Relations extérieures", I love it) is a kind of reaction against earlier French clubs whose whole reason for existence was copying software,

Members would spend three hours of an evening doing nothing else. This kind of activity, (which members who attend our meetings will know is not BASUG practice) is not only illegal but ridiculous. The days of indiscriminate acquisition of material have long since gone - there is only so much that one can assimilate and utilise. What is worse, it lends credence to Howard Stagner's claim that Apple regards user groups as no more than pirating organizations. Sometimes they are.

So, although quite different from BASUG, the MMCC seems to be close to us in ideals. Although new they are already at a point which we are trying to reach - witness our so far abortive attempts to run a PASCAL course. For our part we intend to help them with tips about how to produce a amateur magazine. The type of quality achieved in Hard Core, incidentally, is an important element responsible for the recruitment of overseas members.

We also spoke to Zeljko Lalic of the Yugoslavian Apple Users Group where Apple is fighting an uphill struggle because of foreign currency restrictions. The only way for them to enter the country is in the form of spare parts which are then assembled in Yugoslavia. This drives the price up but the Apple is still in great demand. As for users, these are exclusively businessmen or institutions. "Private individuals" are not allowed to own computers. One wonders how the authorities will succeed in preventing 'guest workers' returning from Germany from owning their own handheld Sharp computers? I hope someone in Yugoslavia will write in to say that I am mistaken and that there is no such law. Zeljko, incidentally, is a living proof of a long held theory of mine that certain people in middle Europe manage to get around the world far more effectively one way or another than inhabitants of the so-called 'free world'.

But what about Holland? Thriving, thriving, some 700 members or so, all very active, very bustling. Apart from the ability to get on their feet and present their case in impressively fluent English (don't worry, BASUG managed to field three Dutch speakers!) members of the Dutch club seem to be hobbyist in orientation, all devising their own programs and very low cost devices (like lower case chips, etc.) made available to members. As a gesture of friendship G. van de Woude presented BASUG with a Super Sector Editor written by Paul Spee of the AGGN (Apple Gebruikers Groep) together with documentation. This is to be made

available to BASUG members for non-commercial use (details to be announced later).

The AUGS group similarly seems to be successful in culling original contributions in software and hardware from its members, and this again is a direction BASUG would want to go in. Not easy.

BASUG's membership would seem to be broader than that of other clubs (I stand to be corrected), covering a wider spectrum of professions and reasons for joining. In other clubs the hobbyist element and spirit seems to be dominant. I personally think we have put our finger on the members' needs all right and are more likely to come with something which would assist, say, the application of microcomputer technology to the study of kidney disease than if we had a more hobbyist orientation.

Wolfgang Dederich assembled an impressive display of European made cards, and I mention only a staggering 16K Z80-driven Printer Buffer Card (Get it? You write 16K's worth of text or whatever to the buffer card which then gets on with printing it, while you do something else with your Apple).

I have to admit that I failed to make contact with the Belgian group, for which I apologise. The only thing I can only say is, please write to Hard Core and tell us about yourselves, tell us what you are up to, anything of interest.

One of the most important insights to emerge from the meeting in Utrecht is that we all need to know: what is going on and to get the lines open. IAC Europe may be in never-never land, but Hard Core is in the here and now. Use it! Write to us with short pieces about your clubs, but also make substantive contributions (on Apple Writer disks please!). Don't worry about writing in fancy English - we can edit out your boo-boos. If you send camera clean hardcopy then follow the column widths in Hard Core exactly.

So who are the silly people who are prepared to take three days out of their lives, and pull a big roll of money out of their pockets all to stand for eight hours at a market stall and not see anything of Utrecht?

Let's take it by the carload: many thanks to Tom Winkel (a native of Utrecht) for driving myself, Stuart Morley and John Molloy to Utrecht and for finding us all a hotel, Richard Teed was rewarded for transporting John Sharp, Derek Botfield, Linda and Richard

Zawadski, by having to watch his Land Rover, complete with BASUG gear, being towed away on the back of a crane truck. (His overnight parking space had been magically transformed by morning into a flower market! He had obviously got his spells wrong). After much gesticulation and the exchange of banknotes the driver agreed to release the vehicle. Dave Bolton had done the sensible thing and flown over, at the same time giving Sarah (his daughter) a birthday gift (her first flight and a day trip to Amsterdam). She returned the compliment by drawing the names of three BASUG members in the raffle for redundant Apple cards in the evening. We have to assume it was coincidental. Fran Teo and business partner Brian Jolly also spent most of the day on their feet at our stand. The thanks of the committee go to all these crazy people and we hope they derived some benefit from it too. Lastly our thanks go to Wolfgang Dederichs for putting the afternoon conference together so successfully in the face of considerably odds. But please, Wolfgang, next time you want us to give you a hand stapling a thousand leaflets together, try not to make it midnight. We have enough of that with Hard Core.

A DISKOURSE, or "FLOPPY FOLLIES", or WITSMILTCH

By Frank Kay

I am sure that many readers accept that they may reliably buy floppy disks for their Apples by simply asking for exactly that kind, and have no interest in why there are differences between these and diskettes for any other drive which uses the 5 1/4" format. However, there has been quite a lot of correspondence from members describing their experiences in having acquired diskettes which for some reason don't seem to work in their Apple drives, so a brief description of the various terms that apply might be useful.

Firstly, a floppy "straight from the box" (unless you have paid extra for a pre-formatted one!) is completely blank magnetically, and a diskette controller will not be able to decipher any meaningful information from it - hence the buzz, followed by "I/O ERROR", that emanates from your Apple if you put a new diskette in and try to read it!

A diskette in fact contains two types of information. It is easy to think of the first type, which is in fact the data that you ask your Apple to write to the diskette so that you may recall it at some point in the future. The second type is not so obvious, and may be thought of as the "pigeon holes" that your data is kept in. A "pigeon hole" is a sector on the diskette, and the diskette system needs to be able to identify each "pigeon hole" so that it

can in fact do what you expect of it. "pigeon holes" are generally referred to by specifying the side, track, and sector of the diskette that each one occupies, and it is the way in which this information is handled that determines the description of a particular diskette type. Most diskette operating systems provide programs that allow a diskette to be "formatted", or, in other words, to have all its "pigeon holes" set up with blank data in the format that it understands. In the Apple case, this happens when you "INIT" a diskette.

Sidedness is simple. A diskette generally has two sides (sic), but a single-sided diskette means that the surface is only warranted to be good on the side that is conventionally used by a single-sided drive. The single side is actually the one on the opposite side of the diskette from the manufacturer's label. In fact, most manufacturers nowadays set out to make double-sided diskettes, and sell those which have emerged from the factory with only one good side as single-sided. If production quality is too good, they will make up the requirement for single-sided disks by selling some which in fact have two good sides as single-sided, thereby losing some money! But be warned - you cannot tell, except by trying each diskette individually, whether the diskette is good both sides. Taking this on trust is asking for data to be lost! Double sided diskettes are most commonly used in drives which have a read/write head for each side (the Apple drive doesn't), but may also be "flipped" (in other words taken out of the drive and turned over). This latter trick requires that you cut an appropriate write-permit notch in the diskette cover.

Now, a word about recording density. You will find diskettes are sold as single or double density, and are now becoming available certified for 80 recording tracks (the Apple drive only uses 35). The greater the recording density, the more difficult the diskette surface is to make, and so the more expensive they are. Diskettes certified for use at higher densities will certainly work at lower ones. The reverse is not true, however, and if you try it, you will reap similar rewards to those of using both sides of a single-sided diskette.

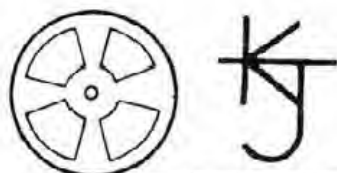
Next, the sectoring technique. In the early days of diskette technology, provision was made for the diskette controller to determine where the sectors were on a diskette track by sensing the presence of a small hole in the diskette, one for each sector. There is usually an extra hole to indicate the first sector, and this technique is called hard-sectoring. By using this technique the diskette controller can locate the data on the diskette without needing to use the read/write circuitry, which can save processing time. It is still necessary to have formatting information around the data for reliability. Since most microprocessor applications have nothing to do but wait for the data when it is required from the diskette, and because the controller is simpler to build that way, the technique of soft-sectoring has become common. In this, the presence of a sector is determined by reading the formatting information continuously until the required sector is found. This formatting information is magnetically stored on the diskette, rather than physically indicated by a sensing hole - hence the name. A first-sector hole is usually

present on the diskette, since this may be useful to the controller. There is no physical difference between hard- and soft-sectored diskettes other than in the number of sectoring holes that are present. For the curious, you can see the sensing holes on a diskette - but be careful, since fingermarks on the recording surface will not improve the readability of data. If you hold the diskette carefully by one corner of its cover (that's the square black bit!), label side up, and gently grip the inside edge of the big hole in the middle of the magnetic surface (that's the grey/brown bit!) with thumb and forefinger of the other hand, you can turn the magnetic diskette. As you do this, within one revolution you will see a small hole in the magnetic diskette through the small hole in the diskette cover - if only one is seen, the diskette is soft sectoring, the usual standard for Apple (it's cheaper). Don't worry if there is more than one, though - simply read on, once you have put the diskette back in its envelope (which you always do, don't you?).

Finally, what use is all this information? I hope it will reassure Apple owners, since, in fact, the Apple diskette controller and drive requires only single-sided, single-density diskettes, and does not care whether the sectoring is hard or soft, since the Apple drive does not even have a sector-hole sensor! So you can use absolutely

any 5 1/4" diskette, and need only discriminate on cost. This latter point is important to all of us, and the only comment that need be made is that the very cheapest diskettes, when similar formats are compared, will not hold data reliably for as long as more costly ones. It is a matter for the user to decide, whether long-term reliability is worth paying for. Obviously, for business use, it is of paramount importance, and it is not worth saving a few pennies if you then have to spend time (and money) to sort out problems resulting from data errors. There is an add-on for cheaper (thinner!) diskettes, called a reinforcing ring, which can be applied for durability to the big hole in the middle of the diskette. This also improves the grip of the diskette drive on the diskette, and can improve reliability that way. It is possible to buy diskettes with these rings already in place, too. For the hobbyist, the choice of diskette is determined much more by his pocketbook, and I should perhaps end by reminding everyone that BASUG offers very good value in diskettes to members - you can possibly buy cheaper elsewhere, but the quality you will have to prove for yourselves!

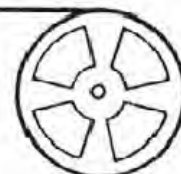
* (Why Is There So Much In Life To Confuse Humans?)



kay-jay studios

135 COLES GREEN ROAD LONDON NW2 7HN

TEL: 01 450 6993



COMPUTERS
MONITORS
PAPER
FINANCE
DISKETTES
CASSETTES
PRINTERS
SOFTWARE
BUGS
PLASTIC MONEY
PRICE LIST
PERSONAL SERVICE

BY APPLE
BY KAGA
BY THE PROUSAND
BY BOWMAKER LTD
LY PAST
BY AUDIO MEDIA
BY MASON & QUKE
BY HANY
BY MISTAKE
BY ACCESS/BARCLAYCARD
BY REQUEST
BY KEN JONES & HIS CREW

APPLESOFT "INPUT ANYTHING" ROUTINE UPDATED

By Ian Trackman

Since my article in Volume 1, Number 1 of Hardcore, I have been finding ways of improving the original machine code sub-routine, which allows commas, colons, semi-colons and quotation marks to be entered in response to Applesoft Basic's INPUT command. I am grateful to Peter Meyer of California (the author of "Agenda Files" in the Apple Special Delivery Catalog) for a number of helpful comments and suggestions.

One point that I did not mention explicitly in my first article (and which still applies to the revised sub-routine) is that if your string INPUT now accepts commas as part of the input string itself, you cannot use a comma as a multiple entry delimiter, i.e. INPUT A\$, B\$ will not work.

Four improvements have now been made to the original sub-routine :-

- a) the sub-routine will recognise Control C as a Break command
- b) a check is made that the variable name is a valid string variable
- c) the variable name is assigned directly by the sub-routine, rather than needing a MID\$ conversion
- d) it is not necessary to assign a dummy first string variable.

The revised sub-routine uses eight calls to Applesoft Basic's internal routines and so cannot be used (without conversion) with Applesoft in Ram at \$800.

The sub-routine (which is re-locatable) takes up 70 bytes and can conveniently be POKE'd into memory from Basic at locations 768 to 837 (\$300 to \$345) as follows :-

```
10 FOR I = 768 TO 837
20 READ X
30 POKE I, X
40 NEXT
50 DATA 169, 58, 32, 192, 222, 169, 132, 32, 192,
222, 32, 227, 223, 36, 17, 16, 43, 162, 0, 32, 117,
253, 173, 0, 2, 201, 131, 240, 36, 138, 72, 32, 57,
213, 104, 72, 32, 82, 228, 162, 0, 160, 2, 32, 226,
229, 160, 2, 185, 110, 0, 145, 131, 136, 208, 248,
104, 145, 131, 96, 162, 163, 76, 18, 212, 41, 127,
76, 99, 216
```

The syntax of the sub-routine is :-

```
CALL nnn : INPUT xx$
```

where nnn is the address at which you have stored the start of the sub-routine (usually 768). As a matter of good programming style, nnn could also be a variable name e.g. INPT with the value of that address. CALL nnn must be followed by a colon and the "INPUT" command. However, both the colon and the INPUT command are dummies used to make your program more readable and can be changed, or even removed, by suitable amendments to the first ten bytes of the machine-code of the sub-routine. Finally, xx is the name of the simple or array (e.g. IN\$(4)) string variable where the response to the INPUT is to be stored.

Here is the annotated assembly language source code of the sub-routine :-

```
SOURCE FILE: STRING INPUT 2
0000:      1 *****
0000:      2 *
0000:      3 *STRING "INPUT ANYTHING" ROUTINE
0000:      4 *
0000:      5 *VERSION 2
0000:      6 *
0000:      7 *COPYRIGHT (C) 1981 IAN TRACKMAN
0000:      8 *
0000:      9 *****
0000:     10 *
0000:     11 *INPUTS ONE BASIC STRING
0000:     12 *ACCEPTS COMMA, COLON, SEMI-COLON & QUOTES
0000:     13 *WITHOUT "EXTRA IGNORED" ERROR
0000:     14 *
0000:     15 *****
0000:     16 *
0011:     17 VALTYP EQU $11      ;FLAG TO INDICATE NUMBER OR STRING VARIABLE
006F:     18 FRETOP EQU $6F   ;BOTTOM OF STRING STORAGE
0083:     19 VARPNT EQU $83     ;USED BY PTRGET TO POINT TO VARIABLE'S ADDRESS
0000:     20 *
0200:     21 LINBUF EQU $200    ;START OF KEYBOARD INPUT BUFFER
```

```

D412:      22 ERROR EQU $D412
D539:      23 GDBUFS EQU $D539
D863:      24 BREAK EQU $D863
DEC0:      25 SYNCHR EQU $DEC0
DFE3:      26 PTRGET EQU $DFE3
E452:      27 GETSPA EQU $E452
E5E2:      28 MOVSTR EQU $E5E2
FD75:      29 NXTCHAR EQU $FD75
0000:      30 *
----- NEXT OBJECT FILE NAME IS STRING INPUT 2.OBJ0
0300:      31 ORG $300
0300:      32 *
0300:      33 XXXXXXXXXXXXXXXXXXXXXXXXXXXX
0300:      34 *
0300:A9 3A 35 LDA #$3A ;CHECK FOR ";"
0302:20 C0 DE 36 JSR SYNCHR ;COMPARES NEXT BYTE WITH ACCUMULATOR, ERROR IF MISMATCH
0305:A9 84 37 LDA #$84 ;CHECK FOR "INPUT"
0307:20 C0 DE 38 JSR SYNCHR
030A:20 E3 DF 39 JSR PTRGET ;GET ADDRESS OF NEXT NAMED VARIABLE
030D:24 11 40 BIT VALTYP ;IS IT A STRING VARIABLE ?
030F:10 2B 41 BPL EXIT1 ;ERROR IF NOT
0311:      42 ;
0311:A2 00 43 LDX #0 ;READY FOR NXTCHAR (X IS THE OFFSET FROM $200)
0313:20 75 FD 44 JSR NXTCHAR ;INPUT A STRING INTO KEYBOARD BUFFER
0316:AD 00 02 45 LDA LINBUF ;FIRST CHARACTER
0319:C9 83 46 CMP #$83 ;IS IT CONTROL C ?
031B:F0 24 47 BEQ EXIT2 ;BREAK IF IT IS
031D:      48 ;
031D:8A 49 TXA ;SAVE LENGTH OF INPUT STRING (IN X) ON STACK
031E:48 50 PHA
031F:20 39 05 51 JSR GDBUFS ;CLEAR HIGH BITS OF CHARACTERS
0322:68 52 PLA ;RECOVER LENGTH
0323:48 53 PHA ;& SAVE IT AGAIN
0324:20 52 E4 54 JSR GETSPA ;GET SPACE IN STRING STORAGE AREA
0327:A2 02 55 LDX #<LINBUF ;ADDRESS LO
0329:A0 00 56 LDY #>LINBUF ;ADDRESS HI
032B:20 E2 E5 57 JSR MOVSTR ;AND MOVE THE STRING THERE
032E:A0 02 58 LDY #2 ;LOOP COUNTER
0330:B9 6E 00 59 LOOP LDA FRETOP-1,Y ;-1 FOR Y OFFSET
0333:91 83 60 STA (VARPNT),Y ;STORE LOW & HIGH ADDRESS POINTERS
0335:88 61 DEY
0336:D0 FB 62 BNE LOOP ;BOTH STORED ?
0338:      63 ;
0338:68 64 PLA ;RECOVER LENGTH
0339:91 83 65 STA (VARPNT),Y ;& STORE IT
033B:60 66 RTS ;BACK TO BASIC
033C:      67 ;
033C:A2 A3 68 EXIT1 LDX #$A3 ;"MISMATCH ERROR"
033E:4C 12 04 69 JMP ERROR ;BASIC'S ERROR HANDLER
0341:      70 ;
0341:29 7F 71 EXIT2 AND #$7F ;CLEAR HIGH BIT
0343:4C 63 D8 72 JMP BREAK ;BASIC'S BREAK HANDLER
0346:      73 *
0346:      74 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

11 VALTYP	6F FRETOP	83 VARPNT	0200 LINBUF
0330 LOOP	033C EXIT1	0341 EXIT2	D412 ERROR
D539 GDBUFS	D863 BREAK	DEC0 SYNCHR	DFE3 PTRGET
E452 GETSPA	E5E2 MOVSTR	FD75 NXTCHAR	
D863 BREAK	D412 ERROR	033C EXIT1	0341 EXIT2
6F FRETOP	D539 GDBUFS	E452 GETSPA	0200 LINBUF
0330 LOOP	E5E2 MOVSTR	FD75 NXTCHAR	DFE3 PTRGET
DEC0 SYNCHR	11 VALTYP	83 VARPNT	

A SPECIAL OFFER TO HARDCORE READERS!

20% OFF
Apple PROGRAMMING
UTILITIES

'SUPER HELPFUL' APPLE PROGRAMMING UTILITIES

Aids 30

A set of useful extensions to the usual DOS and Applesoft commands, these include RENUMBER, AUTO, APPEND, and MERGE, DUMP (to give the value of all simple variables), HELP (to identify location of error), RECOVER and others. A very useful aid in writing and debugging Applesoft programs.

DOS 3.2 or 3.3 VERSIONS (Please Specify) **£40**

Convert II

Converts Integer Basic programs to Applesoft, even if you don't have Integer onboard. (Warning—will not accommodate embedded machine code). Convert-II will either provide a 're-type' under Applesoft or attempt true conversion indicating problem areas.

DOS 3.2 VERSION **£40**

Disc IX

Allows you to carry out operations normally difficult or impossible under DOS, for example renaming the greeting programs, sorting catalogue into alphabetical order, recovering deleted files if possible, finding embedded control characters in file names and inspecting individual track sectors.

DOS 3.2 or 3.3 VERSIONS (Please Specify) **£40**

Edit 3

A line based editor for Applesoft basic. An individual tool for all Applesoft programmers. Features include FIND, INSERT, CHANGE, and DELETE, pausable listing and other facilities. Edit-3 is a machine code program similar in operation to many mainframe editors.

DOS 3.2 or 3.3 VERSIONS (Please Specify) **£40**

Password 69

A simple but effective way to protect disks from unauthorised users.

DOS 3.2 or 3.3 VERSIONS (Please Specify) **£40**

String 16

Adds seventeen string functions to those available under Applesoft. Includes fast string array sort (1000 strings in less than 10 seconds), matching and replacement functions and several general string manipulation functions.

DOS 3.2 or 3.3 CONVERSIONS (Please Specify) **£40**

Symbol 77

Allows you to plot text and other symbols, including user defined ones on the high resolution screens. Also scrolling, partial clearing, inverted and rotated symbols. Machine code program may be relocated, complete with example program and easy to use symbol generators software.

DOS 3.2 or 3.3 VERSIONS (Please Specify) **£40**

Boot II

This program allows you to update a standard DOS 3.2 disk to boot up either 13 or 16 sector disk drives, leaving you in the normal DOS 3.2 environment.

Also contains a utility program, Boot-16, which enables you to boot a DOS 3.3 disk on a 13 sector controller. **£30**

S-C Assembler

This disk based assembler/editor enables you to create machine code programs using standard 6502 Op-Codes, as well as the Sweet-16 psuedo machine codes designed by Steve Wozniak. True macros may be assembled from disk. Assembly may be made directly to disk enabling large programs to be written to reside anywhere in memory. **£55**

Double Precision Arithmetic

21 Digit precision for Applesoft (Written in machine code by the authors of the S-C Assembler)

DOS 3.2 or 3.3 CONVERSIONS (Please Specify) **£55**

Memory Manager

Memory Manager allows you to relocate your disk operating system on a 16K RAM card. This frees the memory for your Applesoft programs and its variables. It allows you to run standard Applesoft programs that do not use the CHAIN function.

DOS 3.3 VERSION **£30**

Personal
Computers
Limited

194-200 Bishopsgate London EC2M 4NR Tel. 01 626 8121

To: Personal Computers Limited, 194-200 Bishopsgate, London EC2.

Please send me I enclose a cheque for £

Name Position

COMPANY NAME ADDRESS

S.B.D. SOFTWARE

15, Jocelyn Road, Richmond TW9 2TJ.
Tel: 01-948 0461. Telex: 22861

PURCHASE YOUR COMPLETE APPLE II COMPUTER SYSTEM FOR THE LOWEST PRICE IN THE U.K.

- 1 APPLE II EURO PLUS 48K
- 1 DISK DRIVE WITH CONTROLLER
- 1 DISK DRIVE WITHOUT CONTROLLER
- 1 HITACHI 10" MONITOR
- 1 EPSON MX-80FT PRINTER + INTERFACE
- 1 MAGIC WINDOW WORD PROCESSOR
- 1 BASIC MAILER
- 1 VISICALC 3.3
- 1 DAN PAYMAR LOWER CASE ADAPTOR
- 1 BOX OF DISKETTES

ALL ITEMS GUARANTEED 1 YEAR
ALL MANUALS INCLUDED
LIMITED QUANTITIES AVAILABLE! HURRY!

ALL FOR ONLY £1,975.00
CREDIT CARD SALES ADD 3%

BUSINESS SOFTWARE

MAGIC WINDOW, BASIC MAILER, DAN PAYMAR - All 3 for **£145.00.**

MAGIC WINDOW will instantly convert your Apple system into a word-processor, no modification or fancy gadgets to buy. Magic Window's 4-way scrolling allows you to type up to 80 CHARACTERS per line, will show you exactly how your letter will be printed. Inserting, deleting, centering, you can see it all on the screen. **£79.95**

BASIC MAILER is a mailing list merge system design to take **MAGIC WINDOW** documents, files and replace names, addresses or any other sections of the document with individual data, creating customised letters, invoices, etc. **£49.95**

DAN PAYMAR lower case adaptor. **£39.95**

VISICALC 3.3 THE UPGRADED VERSION **£99.00**

VISIDEX A most useful cross-reference of information. **£99.00**

VISILOT Plot your data onto high-res. graphs. **£75.00**

BRAIN SURGEON Thoroughly test your Apple II. **£30.00**

D.B. MASTER Computed fields, Statistical Analysis **£130.00**

UTILITIES

EXPEDITER II At last you can compile your Applesoft programs into machine code and watch it run 2-20 times faster. **£75.00**

CRAE Co-Resident Applesoft Editor. Anyone writing software on the apple needs this editor. **£19.95**

CRAE & MCAT **£29.95**

SUPER DISK COPY III The most versatile copy program on the market. Initialize a diskette with or without DOS sectors. Copy files one by one or the entire disk. View the catalog and then see a display of the diskettes free and used sectors. Copy DOS 3.2 to 3.3 and visa-versa, many other commands. **£24.95**

APPLEGUARD Protect against bit copiers, protect your software. **£200.00**

AOPT - APPLESOFT OPTIMIZER, remove REM's. Pack as many instructions as possible per line. **£19.95**

APLUS - STRUCTURED BASIC. Write programs in a structured manner with your new additional commands and then compile into a regular APPLESOFT program. **£19.95**

DOS PLUS. Three new DOS Commands built-in 5 commands are user-definable. You can now FLIP easily between DOS 3.3/3.2 from within the program. Also DOS Command Editor. Edit the names of the DOS commands and initialize disks with your own DOS. **£19.95**

DISK RECOVERY Scan your disks and mark faulty tracks so they are not used. Also able to REDO VTOC which may re-cover your messed-up disk. **£24.94**

BACK-IT-UP Bit copier to back-up your protected software. **£50.00**

ARCADE GAMES

Space Warrior	£12.95	Star Mines	£14.95
Alien Rain	£12.95	Apple Panic	£14.95
Snoggle	£12.95	Sneakers	£17.95
Demon Derby	£12.95	Raster Blaster	£17.95
Galaxy Wars	£12.95	Star Thief	£17.95
Gobbler	£12.95	Space Quakes	£17.95
Star Cruiser	£12.95	Missile Defence	£17.95
Alien Typhoon	£13.95	Pegasus II	£17.95
Space Eggs	£14.95	Threshold	£19.95
Autobahn	£14.95	Epoch	£19.95

ADVENTURE GAMES

Mission Asteroid	£9.95	Ulysses & the golden fleece	£17.95
Mystery House	£12.95	Oldorf's Revenge	£11.95
The Wizard & The Princess	£17.95	Tartarian	£14.95
Cranston Manor	£17.95	Creature Venture	£14.95
Soft Porn Adventure	£17.95	Wizardry	£29.45

CARD & SPORT GAMES

German Whist	£9.95	Cribbage	£12.95
International Grand-Prix	£18.95	Pool 1.5	£18.95
Draw Poker	£14.95	Hi-Res Soccer	£22.95

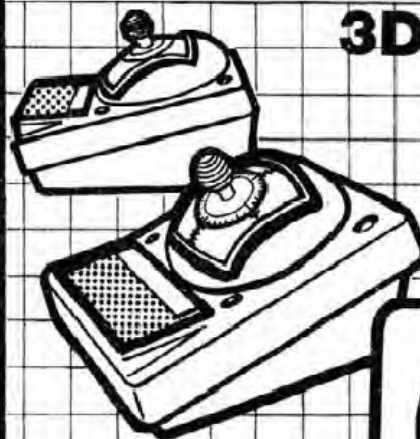
STRATEGY GAMES

Galactic Empire	£12.95	Tawala's Last Redoubt	£14.95
Galactic Trader	£12.95	Golden Mountain	£9.95
Galactic Revolution	£12.95	Kubic	£9.95

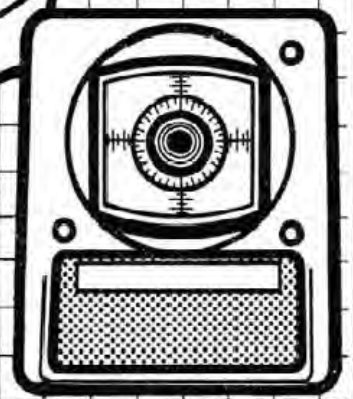
Add 15% VAT. Postage and Packing Free.

Dealer enquiries welcome.

Write or phone for full catalogue of available software.

ROBO
com**3D BIT
STIK****BASUG
SPECIAL OFFER**

AT LAST—A HIGH
PRECISION
'COMBINATION'
JOYSTICK DESIGNED
SPECIFICALLY FOR
YOUR APPLE.



Superb mechanical and hardware features make this the most versatile and accurate analogue controller available.

FEATURES INCLUDE:

- A tracking hemisphere moves about 3 axis enabling interactive program control not previously possible.
 - Built in electronic interfacing reads potentiometers via potential divider circuitry offering 10 times better linearity than ordinary variable resistors.
 - 6 integrated trimmers allow all axis to be trimmed to optimum resolution with no dead boundary areas.
 - Unique direct drive linkage with built in counterbalancing eliminates all control slack and enables fast positive setting.
 - State of the art low-torque plastic track potentiometers provide noise free operation over 5 times the life of ordinary potentiometers.
 - Ergonomic case design with convenience features such as rubber 'eyelid' bumper, comfortable palm mat, non-slip feet, positional scaling and extra long lead.
 - Software support program including full feature C.A.D. package available Mid-January.
 - To encourage the development of BIT STIK interactive software just 30 of these units are available to BASUG members at only **£69.50** + VAT + P. & P. (Total **£82.50**).
 - Software competition details are included.
- Cash refund if not delighted and returned within 10 days.

Make cheques or P.O.s. payable to:

ROBOCOM LIMITED

**CIL BUILDING
GOODWIN STREET
LONDON N4 3HQ
01-263 3388**

BIT STIK is designed and made in England by ROBOCOM LTD., offer applies to first 30 orders received. Telephone to confirm availability before ordering.

Verbatim®



MINIDISKS

Verbatim, the world's leading producer of minidisks, makes the widest range of minidisks available for word processing and data processing applications. Intent on maintaining this position of leadership, we constantly strive to make our minidisks even better.

Each Verbatim minidisk is equipped with a hub reinforcing ring to aid in registration, reduce errors and give better alignment repeatability.

**Try new Verbatim disks
for the performance
of a lifetime.**

SPECIAL OFFER TO B.A.S.U.G. MEMBERS ONLY

10 x MINIDISKETTES **£19.50** INCLUSIVE OF V.A.T. AND POST.

I ENCLOSE A CHEQUE MADE PAYABLE TO B.A.S.U.G. CHEQUE No PLEASE FORWARD TO ME BY RETURN AT THE ENCLOSED ADDRESS.

NAME

ADDRESS

..... POST CODE

THE DOS COMMAND SYNTAX TABLE

By Cliff Wootton

Have you ever wondered how DOS knows whether a particular command is allowed to be executed in immediate mode, or what parameters are required with the various commands. Well - hidden away in the middle of the DOS is a table which has an entry for each command in the DOS which defines just that. This table was modified from DOS 3.1 to DOS 3.2 but has remained unchanged since then. That explains why certain of the text file commands can no longer be executed from the keyboard.

Figure 1 shows a hex dump of this area of the object code for the DOS.

```

A909: 21 70 A0 70 A1 70 A0 70
A911: 20 70 20 70 20 70 20 70
A919: 60 00 22 06 20 74 22 06
A921: 22 04 23 78 32 70 30 70
A929: 40 70 40 80 40 80 08 00
A931: 08 00 04 00 40 70 40 00
A939: 21 79 20 71 20 71 20 70

```

Figure 1 A hex dump of the command syntax table

This table can be reformatted into pairs of bytes each corresponding to a particular command. They are listed in this fashion in Figure 2.

```

INIT      : 21 70
LOAD      : A0 70
SAVE      : A1 70
RUN       : A0 70
CHAIN     : 20 70
DELETE    : 20 70
LOCK      : 20 70
UNLOCK    : 20 70
CLOSE     : 60 00
READ      : 22 06
EXEC      : 20 74
WRITE     : 22 06
POSITION  : 22 04
OPEN      : 23 78
APPEND    : 22 70
RENAME    : 30 70
CATALOG   : 40 70
MON       : 40 80
NOMON     : 40 80
PR #      : 08 00
IN #      : 08 00
MAXFILES  : 04 00
FP        : 40 70
INT       : 40 00
BSAVE     : 21 79
BLOAD     : 20 71
BRUN      : 20 71
VERIFY    : 20 70

```

Figure 2 Syntax table disassembled by command

A pattern is beginning to emerge here showing particular values cropping up more often than others. The pattern is not wholly clear from the format of Figure 2 so Figure 3 is replotted using a binary notation. Each digit is numbered at the top of the columns.

COMMAND	BASE ADDR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
INIT	A909	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0
LOAD	A90B	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
SAVE	A90D	1	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0
RUN	A90F	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
CHAIN	A911	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
DELETE	A913	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
LOCK	A915	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
UNLOCK	A917	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0
CLOSE	A919	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
READ	A91B	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0
EXEC	A91D	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0
WRITE	A91F	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0
POSITION	A921	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0
OPEN	A923	0	0	1	0	0	0	1	1	0	1	1	1	1	0	0	0
APPEND	A925	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	0
RENAME	A927	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	0
CATALOG	A929	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0
MON	A92B	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
NOMON	A92D	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
PR #	A92F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
IN #	A931	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
MAXFILES	A933	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
FP	A935	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0
INT	A937	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BSAVE	A939	0	0	1	0	0	0	0	1	0	1	1	1	1	0	0	1
BLOAD	A93B	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1
BRUN	A93D	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1
VERIFY	A93F	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0

Figure 3 Binary disassembly of the command table

Now we are getting somewhere; let's see what commands have a "1" entered beside them in the various columns.

Column 1

The only commands that have a 1 in this column are LOAD, SAVE and RUN. What have they got in common? They are all basic commands that will have parameters associated with them when used by DOS but will have no parameters entered when used by BASIC. If the operand is omitted they will be passed to BASIC.

Column 2

This column has a 1 entered for the following commands:

CLOSE
CATALOG
MON
NOMON
FP
INT

All of these are "DOS only" commands that are allowed to be used with no associated parameters.

Column 3

This column has a 1 entered for the following commands:

INIT	CHAIN	CLOSE	POSITION	BSAVE
LOAD	DELETE	READ	OPEN	BLOAD
SAVE	LOCK	EXEC	APPEND	BRUN
RUN	UNLOCK	WRITE	RENAME	VERIFY

The only ones excluded are those that do not require any file names to be attached.

Column 4

This is only entered against RENAME which is the only time that filename 2 is required.

Column 5

This column is only entered for PR # and IN #. The only parameter associated with them is a numeric between 0 and 7.

Column 6

Column 6 is only set for the MAXFILES command. Like PR # and IN #, MAXFILES require a numeric argument but the range is different. In this case it is allowed to be between 1 and 16.

Column 7

Column 7 is set for the following commands:

READ
WRITE
POSITION
OPEN
APPEND

These commands relate to text files and are only allowed to be used in deferred mode. That is, while a program is running.

Column 8

Column 8 is set for the following commands:

INIT
SAVE
OPEN
BSAVE

Column 9

This column is set for the following commands:

MON
NOMON

These two commands are the only ones that make use of the optional C.I. and O parameters.

Column 10

Column 10 is used by the following commands:

INIT	CHAIN	EXEC	CATALOG	BRUN
LOAD	DELETE	OPEN	FP	VERIFY
SAVE	LOCK	APPEND	BSAVE	
RUN	UNLOCK	RENAME	BLOAD	

These can all make use of the optional volume number specification. I think this might have been allowed in the READ, WRITE and POSITION commands although it may have been left out to allow programs to transcend the boundaries between different physical devices.

Column 11

This is very similar to Column 10 in that it applies to the following commands:

INIT	CHAIN	EXEC	CATALOG	BRUN
LOAD	DELETE	OPEN	FP	VERIFY
SAVE	LOCK	APPEND	BSAVE	
RUN	UNLOCK	RENAME	BLOAD	

Obviously parameter 11 applies to something similar to the option volume specification; probably drive.

Column 12

This column is identical to 10 and 11 and therefore probably relates to the slot parameter.

Column 13

Column 13 is set for the following commands:

OPEN
BSAVE

These both make use of the length specifier.

Column 14

This is set for the READ, EXEC and WRITE commands. It applies to the RECORD number that is to be accessed.

Column 15

This column is set for the following verbs:

READ
WRITE
POSITION

These are the only commands that might use the Byte specifier.

Column 16



This column is also set for the machine code commands BSAVE, BLOAD and BRUN. These are the only commands that use the Address Specifier.

After a little time spent experimenting I came up with the table shown in Figure 4

Column Number	Function
(1)	If no parameters go to BASIC
(2)	If no parameters go and Do it
(3)	File name 1 required
(4)	File name 2 required
(5)	Numeric value 0-7 required
(6)	Numeric value 1-16 required
(7)	Use in deferred mode only
(8)	Permission to create a file
(9)	Optional C.I.O. allowed
(10)	Optional V allowed
(11)	Optional D allowed
(12)	Optional S allowed
(13)	Optional L allowed
(14)	Optional R allowed
(15)	Optional B allowed
(16)	Optional A allowed

Figure 4 Column Functions

What are the conclusions to be drawn from this command syntax table? Well, it would be fairly simple to modify the text commands, for instance, to allow an OPEN to be performed at the keyboard. Or we could disallow the immediate mode altogether. This would stop DOS being used if the user's program halted for any reason. We can disable the check for a volume slot, drive, or any other parameter just as easily. We could allow PR# to have a value between 1 and 16 instead of 0 and 7, not that it would be particularly useful. There are many ways to use this table to make a system pirate-proof. This is a problem I have tried to solve for a long time and have decided that there is no absolute cure. It is like making your house burglar-proof. The best you can do is make the thief take as long as possible to crack the problem. This may perhaps discourage him. Thief-proofing software can never be 100% successful since you must provide a way to start the program yourself. That being the case, there is a way in, even if the thief has to single step his way through. The only program that is truly thief-proof is the one that cannot be run at all!

<h1>5$\frac{1}{4}$" DISCS</h1>														
BASIC														
<h2>PERFECT FOR APPLE II AND APPLE III</h2>														
1 box (10 discs) £16 per box														
10+ boxes £15 per box														
100+ boxes £14 per box														
Post to: SYSTEMICS LTD., 21-23 The Bridge, Harrow, Middlesex HA3 5AG.														
Order form Please send me:														
Qty.	Phone orders: 01-863 0079	Price	VAT	TOTAL										
	Boxes of 10 5 $\frac{1}{4}$ " DISCS													
add £1.00 Delivery				£1.00										
VALUE OF ORDER £														
NAME _____														
ADDRESS _____														
Signed _____ Phone _____														
My Access/Barclaycard number is:														
 <table border="1" style="display: inline-table;"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> 														

pascal pages

THERE'S A LIST OF WHAT THE PASCAL LIBRARIES CONTAIN CURRENTLY:

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
--------------	---------	------	-----------	-------------

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
PAL1:				
SYSTEM.APPLE	32	26-Jul-79	6 Data	FROM ITT:INCLUDES THEIR TURTLEGRAPHICS
SYSTEM.PASCAL	36	4-May-79	38 Data	FOR THE ITT2020 ((NO BARS!))
SYSTEM.MISCINFO	1	4-May-79	74 Data	
SYSTEM.EDITOR	45	29-Jan-79	75 Code	
SYSTEM.FILER	28	24-May-79	120 Code	
SYSTEM.CHARSET	2	14-Jun-79	148 Data	
SYSTEM.SYNTAX	14	18-Apr-79	150 Text	
PAL.DEMO.CODE	7	3-Apr-80	164 Code	
BUTTERFLY.DATA	7	15-Feb-80	171 Data	
SYSTEM.LIBRARY	53	18-Jul-81	181 Data	

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
ATTACH:				
FROM I.A.C.:EXPLAINS APPLE BIOS.				
AS DESCRIBED IN HARDCORE MAY '81				

DOC.6.TEXT	24	4-Feb-81	161 Text
DOC.7.TEXT	24	4-Feb-81	185 Text
DOC.8.TEXT	28	4-Feb-81	209 Text
ADMERG.CODE	2	2-Mar-81	237 Code
ADMERG.TEXT	6	2-Mar-81	239 Text
LIST.TEXT	14	14-Aug-80	245 Text
LIST.CODE	4	14-Aug-80	259 Code
FORTFIX.TEXT	12	3-Mar-81	263 Text
FORTFIX.CODE	4	7-Jan-81	275 Code

11/11 files, 171 unused, 171 in largest

I BELIEVE THE FOLLOWING 5 DISKS HAVE BEEN CONTRIBUTED BY THE UCSD PASCAL USERS GROUP IN THE STATES. WOULD THE REAL CONTRIBUTOR PLEASE TAKE A BOW!

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
--------------	---------	------	-----------	-------------

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
PSCAL01:				
TITLE.CODE	4	9-Dec-80	6 Code	
INTRO.TEXT	26	15-Feb-80	10 Text	INTRO TO UCSD PASCAL USERS GROUP
POLICY.TEXT	18	30-Mar-80	36 Text	
BUGS.TEXT	18	9-Aug-80	54 Text	KNOWN BUGS, FROM I.A.C. MARCH'80
DOS3.2A.TEXT	42	28-Jan-80	72 Text	ALL YOU NEED TO KNOW ABOUT DOS3.2
DOS3.2B.TEXT	42	28-Jan-80	114 Text	BUT WERE AFRAID TO ASK.FROM "ILLINOIS
DOS3.2C.TEXT	14	14-Jul-80	156 Text	MICRO. ABBS" & APPLE ORCHARD
COMBINES.TEXT	8	9-Oct-79	170 Text	COMBINES 2-4 TEXT FILES
COMBINES.CODE	4	5-Apr-80	178 Code	
CHASE.TEXT	22	22-Nov-80	182 Text	GAME: USES TEXT SCREEN
CHASE.CODE	10	22-Nov-80	204 Code	
LIST.TEXT	12	4-Feb-80	214 Text	LISTS TEXT WITH HEADINGS ETC.
LIST.CODE	5	5-Apr-80	226 Code	
INTRINSIC.TEXT	6	7-May-80	231 Text	TO ILLUSTRATE BUILT IN PROCS - FROM
				PAGE 84 "PROBLEM SOLVING USING PASCAL"
UNITS.DOC.TEXT	12	12-Mar-80	237 Text	DOCUMENTATION ABOUT UNITS
TYPESET.TEXT	12	29-Nov-78	249 Text	??ANY CLUES?:COPIES FILES USING
TYPESET.CODE	5	5-Apr-80	261 Code	DIFFERENT FONT'S I THINK(?)
UNIVERSAL.TEXT	12	1-Apr-80	266 Text	A SPEC FOR "A UNIVERSAL I/O UNIT"
< UNUSED >	2		278	
18/18 files, 2 unused, 2 in largest				

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
PSCAL02:				
PILOT.TEXT	12	5-Jul-80	6 Text	
PILOT.CODE	8	5-Jul-80	18 Code	
PILOT.DOC.TEXT	16	5-Jul-80	26 Text	
PRETTY.CODE	11	5-Apr-80	42 Code	
PRETTY.TEXT	36	2-Oct-79	53 Text	
PRETTY.DOC.TEXT	10	14-Feb-80	89 Text	
FORMAT2.TEXT	28	16-Nov-79	99 Text	
BLACKJACK.TEXT	20	27-Nov-80	127 Text	
BLACKJACK.CODE	8	27-Nov-80	147 Code	
FORMAT.TEXT	20	16-Nov-79	155 Text	
FORMAT.DOC.TEXT	30	18-Feb-80	175 Text	
FORMAT.CODE	20	5-Apr-80	205 Code	
FORMAT1.TEXT	34	13-Nov-79	225 Text	
FLOWER.TEXT	6	5-May-80	259 Text	
STARS.TEXT	6	6-May-80	265 Text	
SQUARES.TEXT	4	6-May-80	271 Text	
< UNUSED >	5		275	
16/16 files, 5 unused, 5 in largest				

DISK/PROGRAM	SECTORS	DATE	FILE TYPE	DESCRIPTION
PSCAL03:				
DRAWLINE.TEXT	6	9-Oct-80	6 Text	
DRAWLINE.CODE	2	9-Oct-80	12 Code	
MOTGEN.TEXT	24	8-Aug-80	14 Text	
MOTGEN.CODE	8	1-Aug-80	38 Code	
MUD.GUIDE.TEXT	16	4-Aug-80	46 Text	
MUD.PROC.TEXT	10	4-Aug-80	62 Text	
MUD.PROC.CODE	3	4-Aug-80	72 Code	
MORE.MUD.TEXT	8	14-Aug-80	75 Text	
MUD.TEXT	34	4-Sep-80	83 Text	
MUD.CODE	15	4-Sep-80	117 Code	
STEVE.CODE	2	11-Dec-79	132 Code	
SIEVE.TEXT	4	9-Oct-80	161 Text	
FILE10.TEXT	6	22-Jun-80	134 Text	
FILE10.CODE	3	22-Jun-80	140 Code	
LCM.TEXT	4	3-Nov-79	143 Text	
LCM.CODE	2	3-Nov-79	147 Code	
GCD2.CODE	2	7-Sep-79	149 Code	
GCD2.TEXT	6	7-Sep-79	151 Text	
README.TEXT	4	9-Oct-80	157 Text	
RND.SORT.TEXT	6	10-May-80	165 Text	
RANDOMPLOT.TEXT	4	11-May-80	171 Text	
PRINT.OUT.TEXT	6	14-Dec-80	175 Text	
SHELL.SORT.TEXT	6	27-May-80	181 Text	
FIGURE.TEXT	8	29-May-80	187 Text	
COLOR.BAR.TEXT	12	30-May-80	195 Text	
POLOR.EX.TEXT	4	4-Jun-80	207 Text	
REALSTR.PC.TEXT	6	25-Jun-80	211 Text	
POLORGRAPH.TEXT	12	25-Jun-80	217 Text	
POLORLOOP.TEXT	12	26-Jun-80	229 Text	
EPICYCLOID.TEXT	14	3-Jul-80	241 Text	
HYPQ.THREE.TEXT	14	3-Jul-80	255 Text	
CYL.TEXT	8	4-Jul-80	269 Text	
< UNUSED >	3		277	
32/32 files, 3 unused, 3 in largest				

EDITED BY WARREN AVERY

PILOT INTERPRETER FROM "BYTE".

<- IT'S DOCUMENTATION
FORMATS PASCAL PROGRAM TEXT FILES.

<- IT'S DOCUMENTATION
(PART OF FORMATTER)
TEXT VERSION OF THIS GAME

(MORE FORMATTING)

FROM "PROBLEM SOLVING IN PASCAL"
DITTO (ALL USE TURTLEGRAPHICS)
DITTO

PSCAL04:		
PASCALZAP.TEXT	10 1-Mar-81	6 Text CHANGES BYTES: CALL APPLE JAN'81
PASCALZAP.CODE	6 1-Mar-81	54 Code
PRINTNET.TEXT	24 26-Dec-80	16 Text PACKET NETWORK SIMULATION
PRINTNET.CODE	8 26-Dec-80	64 Code
KINETICART.TEXT	10 15-Feb-81	40 Text KINETIC ART: BYTE NOV '80
KINETICART.CODE	4 15-Feb-81	50 Code
TIGERPROC.TEXT	4 8-Nov-80	60 Text I THINK..TURTLEGRAPHICS ON PAPER TIGER
SAILCALC.TEXT	16 20-Dec-80	72 Text CALCULATES SAIL AREA/DISPLACEMENT
SAILCALC.CODE	11 20-Dec-80	88 Code
CENTRON737.TEXT	6 20-Feb-81	99 Text FILE PRINT FOR CENTRONICS 737 (??)
CENTRON737.CODE	4 20-Feb-81	105 Code
< UNUSED >	171	109
11/11 files, 171 unused, 171 in largest		

FROM "PRINCIPLES OF INTERACTIVE
COMPUTER GRAPHICS":INC.:TURTLEGRAPHICS
"APPLE PASCAL MEMORY UTILITY DELIGHT"
-ALLOWS USER TO EASILY ACCESS AND
-DIRECTLY MANIPULATE MEMORY UNDER THE
-PASCAL OPERATING SYSTEM.

-OUTPUTS PRIME NUMBERS

-AN ATTEMPT TO CREATE A NAME FILE,I
THINK: IT CRASHES EASILY!.
CALCULATES LOWEST COMMON MULTIPLE

CALCULATES GREATEST COMMON DENOMINATOR

DESCRIBES PART OF THIS DISK
SORTS & PLOTS A SEQ OF RANDOM NO'S
SAME WITHOUT THE BUBBLE SORT
ATTEMPTS TO SORT & PRINT RANDOM NO'S
SORTS & DISPLAYS RANDOM NUMBERS
DRAWS N SIDED POLYGON, LINKING POINTS
DRAWS HORIZ. & VERT COLOUR BARS
POLAR GRAPHICS
DITTO (THESE FEW ARE QUITE GOOD IF
(YOUR INTERESTED IN SUCH THINGS!)

PSCAL05: ***ARE THERE ANY GERMAN SPEAKERS OUT THERE? ***		
INTKORR.TEXT	6 17-Apr-80	6 Text "DIESES PROGRAM KORRIGIERT LIBRARIES
INTKORR.CODE	3 17-Apr-80	12 Code "DER NAME DER BIBLIOTHEK IST
		"VOLLSTAENDIG ANZUGEBEN" EH??MONITOR.DATA
MONITOR.TEXT	6 29-Oct-80	23 Text "..LAEST TEILLE DES MONITORS IN DIE
MONITOR.CODE	3 29-Oct-80	29 Code "LANGUAGE CARD UND STARTEN DEN MONITOR
SORTDEMO.TEXT	10 21-Sep-80	32 Text
LINEFEED.TEXT	4 17-Oct-80	42 Text
CROSSREF.CODE	3 8-Jan-80	46 Code
CROSSREF.TEXT	8 8-Jan-80	49 Text ***I'M AFRAID I DON'T SPEAK DEUTSCHE
SORTDEMO.CODE	4 2-Nov-80	57 Code ANY VOLUNTEERS TO DESCRIBE THIS DISK?
LINEFEED.CODE	2 2-Nov-80	61 Code
SYSTEM.CHARSET	2 14-Jun-79	63 Data
SYSTEM.SYNTAX	16 11-Nov-80	65 Text
SYNTAX.TEXT	4 12-Nov-80	81 Text
SYNTAX.CODE	3 12-Nov-80	85 Code
PI.TEXT	6 3-Feb-80	88 Text
HANDI.TEXT	8 12-Nov-80	94 Text
HANDI.CODE	3 12-Nov-80	102 Code
PI.CODE	4 12-Nov-80	105 Code
GENSORT.TEXT	10 20-Sep-80	109 Text
SORTDEMO2.TEXT	4 12-Nov-80	119 Text
SORTDEMO2.CODE	4 12-Nov-80	123 Code
STARTUP.TEXT	6 12-Nov-80	127 Text
GRUSS.TEXT	8 12-Nov-80	133 Text
SYSTEM.STARTUP	4 12-Nov-80	141 Code
< UNUSED >	135	145
25/25 files, 135 unused, 135 in largest		

THE ABOVE DISKS BELONG TO THE "PASCAL SDL" (SOFTWARE DISTRIBUTION LIBRARY)
THIS MEANS THAT THEY'VE BEEN CONTRIBUTED TO BASUG FROM OTHER USER GROUPS
(OR IN THE CASE OF PALI:,FROM ITT).

PASCAL PAGE CONTINUED

NEXT ISSUE WE WILL BE ABLE TO INCLUDE AT LEAST ONE DISK FROM THE "PASCAL CSL"
(CONTRIBUTED SOFTWARE LIBRARY)

THE PROGRAMS ON IT HAVE BEEN CONTRIBUTED BY OTHER BASUG MEMBERS.

WE WELCOME SOFTWARE!

WHEN CONTRIBUTING SOFTWARE PLEASE INCLUDE COMMENTS TO:

IDENTIFY YOU, THE CONTRIBUTOR

THE SOURCE OF THE PROGRAM IF NOT ORIGINAL

A BRIEF DESCRIPTION (SUITABLE FOR THESE PAGES) OF WHAT IT DOES

WE CURRENTLY PREFER TEXT & CODE FILES (IT'S A BORE COMPILING LOT'S OF PROG'S!)

SEPERATE DOCUMENTATION FILES ARE ALSO WELCOME.

THE PASCAL LIBRARIES WILL OPERATE ON THE SAME CREDIT SYSTEM AS THE BASIC
PROGRAM LIBRARIES EXCEPT THAT PASCAL CREDITS ARE NOT INTERCHANGEABLE WITH
BASIC PROGRAM CREDITS.

FOR FURTHER DETAILS ABOUT HOW THE LIBRARIES WORK READ HARDCORE AUGUST 1981

PASCAL TEXT HAS A VALUE OF 1 CREDIT

PASCAL CODE HAS A VALUE OF 1 CREDIT (YES THAT'S RIGHT: 2 CREDITS PER PROGRAM!)

OTHER HINTS FOR PASCAL PROGRAMS:

AVOID THE WORD "AQUA" IT'S AN ITT TURTLEGRAPHICS RESERVED WORD!

ATTEMPT TO CHECK FOR KEYING ERRORS TO AVOID SYSTEM CRASHES

TRY TO INCLUDE A "QUIT" PATH IN EVERY PROGRAM, RETURN TO TEXT IF NECESSARY.

AUTOCRAT COMPUTERS

SPECIAL OFFERS

QUALITY 5.25" DISKS

10 for £15.49+VAT

> EPSON MX80 FT <

>> £349 + VAT <<

Post & Packing Free



**SPECIAL
ARRANGEMENTS ON
APPLE HARDWARE FOR
BASUG MEMBERS**

AUTOCRAT COMPUTERS,
264 Preston Road,
Harrow, Middx., HA3 0PY.

Tel: 01 908 3636

MICROSOURCE

MICROSOURCE is a new concept, specialising in utilities for the APPLE & ITT 2020. Now, because as users ourselves, we know what you need and so are in the unique position of being able to offer expert advice. We can thoroughly recommend the products below, either we use them ourselves, or have had them specially written to fit a need. We will gladly supply more information and answer your specific enquiries.

RAMEX CARDS

The only RAM Extension card for the APPLE which does not involve pulling chips out of the mother-board in combination with your System Master Disk, you can now have the equivalent of an Integer or Appleton ROM card, as well as being ready for PASCAL Com. patible with Z80 card, Visicalc, The MILL 6809 card, etc. etc.

PROGRAM LINE EDITOR

Gives you the editing facilities APPLE left out. You can now insert / delete / find / display normal and control characters in lines without having to copy the whole line. No more spaces in strings or fiddling with the cursor keys. With the added bonus of defining your own commands through the ESC key. Cuts down your mistakes and saves you time.

OMNIFONT

A set of programs which allows you to use your EPSON printer to produce hard copy using ANY of the fonts in the DOS 3.3 TOOLKIT. Use the ones provided or define your own from BASIC or even from APPLEWRITE. Use all the facilities of your EPSON (including enlarge, enhance and underline) with APPLEWRITE.

OMNIDOS

Accesses both 3.2 and 3.3 disks. Automatically Uses no extra memory. Solves all the problems of having to Muffin and DeMuffin files. Also includes utilities to personalise your disks, alter DOS commands and protect your disks from copying.

SCREENWRITER

The program for the APPLE owner wishing to use the computer for advertising purposes. It allows you to set up a changing display on the screen in various character sets and colours, in upper and lower case to advertise or inform. Compatible with Versawriter.

APPLEPLOT CONVERTER

Allows you to print the graphs on an EPSON printer in all modes. (Graphics facilities must be present on the printer).

SUPER-VADERS

A totally new version of the ever popular game.

ABC - Apple Bit Copier

This disk allows you to make back up copies of most disks including the copy protected ones.

VISICALC BACK-UP

Allows you to make a back up copy of your Visicalc disk. Please specify 3.2/3.3 DOS.

RELOCATED INTEGER

Allows you to run any Integer program without the expense of a language card, including those using Hi-res graphics. Mini assembler included. Specify memory size and DOS 3.2/3.3.

AUTO-INDEX

Allows you to make a Master Catalog disk, with fully automatic updating, and comprehensive search and sorting routines. Requires 48K and disk. Specify DOS 3.2/3.3.

GAMES EXPANSION PORT

Forget having to take the lid off every time you want to change paddles for other accessories. Run two sets of paddles.

DISK DRIVE EXPANSION PORT

Do you remove your disk controller cards when you transport your APPLE? Wouldn't it be easier just to unplug them from this handy extension port without removing the lid.

BENEATH APPLE DOS

This book is a goldmine of information on how DOS works, what it does and where all the locations are for the various routines.

DOS-SWITCH

Are you fed up of having to boot a 3.3 disk and use BOOT13 or to use your BASICS disk to boot a 3.2 (13 sector disk) or had trouble with the games you bought in 3.2 DOS which will not boot. The answer is to fit this switch, which will allow you to boot a 3.2 DOS disk directly, or switch back to a 3.3 DOS boot mode. Fits onto the disk controller card. Fitted in minutes. DEALERS please ask for terms.

All prices inclusive of VAT, Postage, etc.

MICROSOURCE

1 Branch Street, Park Street,
St. Albans.
Tel: Park Street (0727) 72917.

THE PERSONAL COMPUTER WORLD FAIR

by Tony Williams

All the omens for this Fair were bad: disaster struck the BASUG stand from all sides. Worst, the non-materialization of Hard Core No.4, announcing our participation in the Fair and containing a voucher giving members a fifty per cent entrance discount. Let us pull a discreet curtain down over this sad episode.

Special exhibitions organizer Eddie Payne made a special exhibition of himself, sitting in a broken down vehicle in St Albans containing all BASUG's Fair props. This on the eve of the opening. Meanwhile back at the Fair, Francis Teo and cohorts were not twiddling their thumbs waiting for Eddie but were getting on with the Get-In.

That for me was one of the lasting impressions of the Fair, the unbidden willingness of BASUG members materialising from all sides to pitch in and say Roll-Up, enrolling new members, answering questions they could answer, fending off or buck-passing ones they couldn't. (Eddie Payne says they were not unbidden, he bade, he bade). Special thanks to David Springle of Conqleton who spent two valuable days of his London Experience on our stand.

Well, what of the Fair? "Much better than last year," said David Ahl, a name not unknown in creative computing circles but occupying a modest place behind his bookstall. Now recovered from his recent illness he looks an enviable twenty-one. "The attendance is much heavier than I have been seeing in the States, and a lot of excitement has been generated by the show. There are a lot of things here we simply have not seen yet back home, I mean new products, particularly the little DAI, which can do a lot, the graphics of course. And the BBC machine is attracting a lot of attention." Most of the Creative Computing stand was bought up by John Sharp and hauled off by the carton load. This was because he hasn't enough to read of an evening.

This reporter as the prospective half owner of a BBC beast would have loved to give the machine at least the once over at the Fair, but unfortunately that would have meant bulldozing aside serried ranks (three deep from 10 a.m. to 7, p.m.) of brawny sixfooters with shoulders like wardrobes all raptly ogling the Langham Place

brainchild. No matter, it would appear that the "BBC" machine was not quite what it claimed to be...

The Sinclair of last year had to take a back seat - well not quite. The ZX80/81 Users Club may have cast envious glances at the quality of our mag, but with their six-thousand membership and rising (BASUG stands at around 600 - and rising) they are obviously flourishing. Their stand was in danger of submerging. I didn't have time to ask, but it could have been distraught members asking how to unstick the ice-cream carton reputedly used as a Heat-Sync. One curiosity. Whatever became of Sharp - you know, the one with Katakana script. The handheld one popped up here and there,, but its big brother, so prominent at the North London Fair, took a raincheck.

So, for that matter did Apple. This is getting to be a bit embarrassing. BASUG exists, at least I thought it did, to help out Apple users with their little problems, not to sell Apples. And yet there we were in the front line, as Eddie Payne explained, telling prospective buyers "Why Apple?". Would we be so kind as to direct them to the Apple stand then? Ah well, cough, cough, they aren't here see. Commodore thought it worth it and took over half the top floor, but if you want an Apple - well there is always Personal Computers Ltd, they are the peculiar ones dressed up in Royal Wedding Footpersons livery, ruffled blouses and stripy trousers and that, but despite this they not averse to selling the odd Apple or two. But no flag was flown by Microsense. Never mind, perhaps the new regime will read it differently.

What was in it for us? Eighty odd new members, about as many as we can handle in one sitting, a welter of Apple users' questions heard and we hope answered, a lot of work, and, perhaps most important, further identification of the solid body of members prepared to step up and give us their time; the committee joins Eddie Payne in thanking Stuart Morley, John Molloy, Frances Teo, Richard Zawadski, Jonathon Shaw, Jeremy Ensor, Kenneth Gordon, Warren Avery, Derek Botfield, Ian Trackman for helping on the stand. For future reference; you help us out, you get in free! And you get a BASUG helpers badge.



**INTERNATIONAL
APPLE CORE**
TM

MEMBER OF THE
INTERNATIONAL APPLE CORE

ITT MONTH BIT ADDRESSING

By Ken Gordon.

The ITT 2020 differs from the Apple in that its high resolution graphics screens have a resolution of 360 x 192 and not 280 x 192. This is achieved by displaying Bit Eight (which is the colour bit on an Apple) and an extra bit, Bit Nine. This has caused many problems for ITT users. One of these problems is how to change bit nine and then to examine it.

Changing Bit Nine

To write to bit nine the program must first save the value of that byte's first eight bits.

E.G. 10 A = PEEK (8193)
or \$1000: LDA \$2001
Then the program must address location \$C05E (HEX)

E.G. 20 POKE -16290,0
or \$1003: STY \$C05E

Now the program can change bit nine to be either low or high as required.

E.G. 30 POKE 8193,128 : REM BIT NINE HIGH
or \$1005: LDX #80
\$1008: STX \$2001

Now, in order to restore the original values of bits one to eight without causing bit nine to change, the program must address location \$C05F (hex).

E.G. 40 POKE -16289,0
or \$100B: STY \$C05F

It is now possible to restore bits one to eight.

E.G. 50 POKE 8193,A
or \$100E: STY \$2001 ;RESTORE BYTE

The same process is used to make bit nine zero. The only difference is that 0 is used instead of 128 (\$80) at line 30.

E.G. 30 POKE 8193,0 : REM BIT NINE LOW
or \$1005: LDY #00

\$1008: STY \$2001

Examining bit nine.

The general principle is that the program first looks at the required byte and then at \$C063 to see if bit nine was high or low. When bit eight of \$C063 appears low then bit nine of the last addressed byte in hires memory was also low. If bit eight of \$C063 is high then bit nine of the last addressed byte in hires memory is high.

E.g

```
$1200: LDA $2001
$1203: LDA $C063
$1206: BPL $1210 ; BIT NINE LOW
$1208: LDA #FF ; BIT NINE HIGH
$120F: STA $00
$1211: RTS
$1212: LDA #00 ; BIT NINE LOW
$1214: STA $00
$1216: RTS
```

or

```
10 A = PEEK (8193)
20 B = PEEK (-16285)
30 IF B < 128 THEN 100 : REM BIT NINE LOW
40 PRINT "BIT NINE IS HIGH"
50 GOTO 70 : REM END
60 PRINT "BIT NINE IS LOW"
70 END
```



**Colour
from
EuroApple
without
'Colour Card'**

**14" Colour Monitors
for Computer Application
Model AM 3781 dedicated to Euro
Apple. £325.**

No 'colour card' required
High Resolution Colour Graphics
80 Character Capability
Full Screen Text Window Display
Green or white text

**SPECIAL TERMS FOR BASUG
MEMBERS**

Add 15% VAT

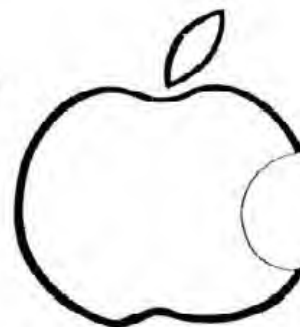
Trade enquiries welcome

PortaTel Conversions Limited
25 Sunbury Cross Centre
Sunbury on Thames Middlesex
Tel: No. Sunbury (09327) 88972
VIDEO MODIFICATION SPECIALISTS



Accutrack Disks . . .

Because data reliability
is the important difference
in disk construction.



Anatomy of a disk.

Flexible disks are simple information storage devices consisting of a magnetic disk enclosed in a semi stiff protective jacket. The disk rotates within the jacket while magnetic recording heads on your data or word processing systems "read" or "write" information on the disk's magnetic surface. Since disk operation is simple, it's relatively easy to make one that works. But building in reliability is something else again. It takes specialized technology to build disks that operate flawlessly over an extended period of time.

What counts in disk construction.

Key design objectives for a disk are listed below. How well a disk measures up to these objectives relates directly to the throughput, accuracy and overall costs for your data or word processing system. No disk measures up better than Accutrack.

- The magnetic coating must be precisely formulated and uniformly applied. Imperfections as small as five millionths of an inch cause signal dropouts, data checks and wasted processing time as well as errors.
- The disk surface must be absolutely clean, totally flat and permanently lubricated to prevent excessive head wear with subsequent signal degradation and eventual loss of information. (This is the most critical objective and the one that's most often compromised

since poor operating results take a while to show up. It's also the area that most affects the long term reliability of your data.)

- The disk must be free to rotate within its jacket without internal drag to avoid further data checks, excessive processing times and errors.
- The jacket must protect the disk from external contamination and damage. It should also remove microscopic particles of debris from the disk surface before they can damage the disk.

Why you'll never find the best disk bargain in the bargain basement.

While there's little apparent difference between other disks and Accutrack, the performance differences can be substantial. Simply stated, an Accutrack disk is premium priced. But the protection it gives your information, the reliability it provides to your operations, and its substantially longer life make it the best disk buy. After all, the real cost of your operations is constructing and processing the data stored on the disk – not the disk itself. It doesn't make sense to trust that data to anything but the best disk. Accutrack.

BASUG BULK BUYING – 500^{off} PRICE FOR 10^s

£17:50 incl FREE LIBRARY CASE !

(ADD 50p POSTAGE / PACKING)

BRITISH APPLE SYSTEMS USER GROUP

P.O. Box 174, Watford WD2 6NF.

WHY ARE 25 MULTINATIONALS C/WP CUSTOMERS?

C/WP is a long established service company based in Rochester Row, London SW1.

C/WP Computers is an Apple authorised level 1 service centre.

C/WP now offers readers of Hardcore the chance to buy at its special prices.

Apple/Visicalc offer

Apple 48K Europlus	£579.00
Disc drive with controller	£300.00
12" green monitor	£140.00
Silentype printer	£180.00
Visicalc 3.3	£100.00
	<hr/>
	£1299.00

VAT and installation extra

C/WP

C/WP Computers
01-828 3127

108 Rochester Row
London SW1P 1JP

Pippin's Page ~~~~

~~~~~  
Edited for younger readers by Vernon Quaintance  
~~~~~

In the last issue we saw how the Apple could be used in Immediate Mode as a sort of large calculator. We also saw how we could PRINT out words and the results of our calculations, if we wanted them. I hope you have done lots of experimenting with the use of commas and semicolons in your PRINT statements, so that you know exactly what each does.

I told you that there was another way of using the computer. This is called Deferred Execution Mode. In this mode we must first write a PROGRAM which the computer will later follow. As you will see, the computer appears to do nothing as we type in the program instructions, unlike the Immediate Mode where it immediately did as it was asked.

Before we can write a new program, we must make sure that there is not part of an old one still in the Apple. To do this type NEW. Do this now. Right, now what exactly is a Program? It is a series of instructions which the computer is to obey, in the correct order, at some later time. How do we make sure that the computer knows the right order for the instructions? We number each line of instructions; this also gives us straight away a difference between Immediate Mode instructions and Program instructions.

Try typing the following short program:-

```
10 PRINT "2 + 6 = ";2+6 <return>
20 END <return>
```

Remember that <return> means press the key marked RETURN. You must do this at the end of each numbered line to tell the Apple that you have finished that line. I will not show this <return> in future, but remember to use it.

Nothing happened as you typed this, did it? That was because each line started with a number and the computer knew that you were writing a program. Now type RUN. As soon as you press the return key your Apple will reply with:-

2 + 6 = 8

Notice the command in line 20. In Applesoft BASIC this is not actually necessary but should always be used in case you want to try your programs on either an Apple with only Integer BASIC, or on some other type of computer. We

have now learned three new commands, and written our first program. Can you remember the new commands? Yes, they are NEW, END and RUN. Go back and read again what each of these does. Now try to write a short program of your own.

Have you noticed that you can keep on RUNning your program over and over again? It is normally still in the Apple's memory until you type NEW or turn off the Apple. Perhaps by now you can no longer see the program on the screen. If you type LIST, look what happens! Your program is there on the screen again. The Apple LISTed out the lines of your program.

Now let us look at another thing we can do in a program, that we cannot do otherwise. Type:-

```
10 PRINT "2 + 6 = ";
20 PRINT 2 + 6
30 END
```

Now RUN this program. The result is the same as before, but notice that the two PRINT statements are on different lines. The PRINT statement in line 10 ends with a semicolon, doesn't it? Remember, this stops the printer or screen from starting a new line there. A semicolon (or comma) used at the end of a line like this is said to be trailing. You will find out as you write programs, just how useful the trailing semicolon can be.

In the short program above, we numbered our lines 10, 20 and 30. We could just as easily have called them 1, 2 and 3 - the program would still have run. It is usual, however, to start our line numbers at either 10 or 100 and then to go up in 10's. The reason for this is that we may want to add some extra lines in the middle. If we use consecutive numbers for our lines, then there is no room for any new lines without re-typing the whole lot. In the program above we could add a new line 15, say, without disturbing the other lines. Try typing something for a line 15, and then LIST your new program. See how the new line has been added in its correct place. BASIC will always sort the lines into ascending numerical order, that is the order in which the commands will later be obeyed.

In all the calculations which we have done so far, the numbers have been directly written in. Suppose we want to use the result of one calculation a little later on in another calculation.

We then need a means of storing the first answer for later use. The way we do this is by means of something called a **VARIABLE**. A variable is like a sort of pigeon-hole, a little box into which we can put a piece of paper with a number on it. We then label the pigeon-hole with a name so that we know where to look for that number later on.

The name which we use for our Variable can be almost anything we like provided three conditions are met. These are:- The name must start with a letter (not a number); The first two characters must be different from the first two characters of any other variable name; The name must not be, or include, any BASIC command word.

A variable is given a value by means of an Assignment statement, using the command **LET**, for example:-

```
10 LET PI = 3.147
20 LET RADIUS = 7
30 LET AREA = PI * RADIUS^2
40 PRINT AREA
50 END
```

Here we have used three variables called **PI**, **AREA** and **RADIUS**. The program directly assigned values to **PI** (in line 10) and to **RADIUS** (in line 20). **AREA** was assigned a value as the result of the calculation in line 30. Type in and RUN the short program above. Now try typing **PRINT PI**, or **PRINT RADIUS**. They are still there, with the values which we gave them, ie 3.147 and 7 respectively. When we use a variable, what we do is like going to the pigeon-hole and looking at the piece of paper, without taking it away.

Now try writing your own program using the commands which we have just learned. Next time we will learn how to give numbers and words to the program as it is running.

Enjoy your Apple programming.

```
100 REM *****
110 REM * HOW TO PUT QUOTES *
120 REM * IN AN EXEC FILE *
130 REM *
140 REM * BY R.D. PURVES *
150 REM *****
160 PRINT : D$ = CHR$ (4)
170 PRINT D$"OPEN EXECDEMO"
180 PRINT D$"WRITE EXECDEMO"
190 PRINT "?" CHR$ (34) "HAPPY " ;
200 PRINT "NEW YEAR" CHR$ (34) ;
210 PRINT D$"CLOSE"
220 PRINT D$"EXEC EXECDEMO"
```

S.B.D. Software is proud to announce their distribution agreement with the most up to date **APPLE-only** magazine in America.

CALL A.P.P.L.E.

Now you can purchase this outstanding magazine for the low price of £1.75 per issue.

Your subscription for 12 or 24 magazines may start from any month in 1981.

Single back issues are available at £2.25 per issue including postage and packing.

A bound volume of the 9 issues in 1980 is available for £20.00 including postage and packaging.

(Please note that in 1980 & 1981 there were only 9 issues published but in 1982 there will be 12 issues.)

CALL A.P.P.L.E. is ahead of any other **APPLE** publication for its technical information and evaluation of all hardware and software for the Apple.

For example:-

In the June 1981 issue **CALL A.P.P.L.E.** had already reviewed the new **APPLE III**.

In the July/August 1981 issue one of the topics in **CALL A.P.P.L.E.** was loading **DOS 3.3** onto the Language Card.

In the September 1981 issue **Word-Processing** reviews have begun, they will continue reviewing 5 different word-processors until they run out of software to evaluate.

SPECIAL INTRODUCTORY OFFER

expires 31st December 1981

12 issues @ £19.50

24 issues @ £39.00

Act Now and Save

NAME

ADDRESS

TOWN

POSTCODE

12 issues Sub. @ £19.50

24 issues Sub. @ £39.00

Please start my subscription

Month

Year

Barclaycard/Access Number

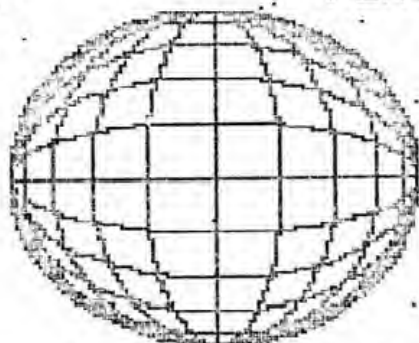
Expiry Date

Please make cheques payable to **CALL APPLE (UK)**

Send to:- **CALL APPLE (UK)**
c/o SBD Software
FREEPOST
RICHMOND
SURREY TW9 1BR
(No postage stamp required)
Telephone: 01-940 5194

BUSINESS BASIC HAS LANDED !

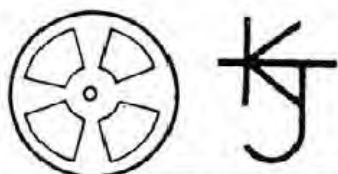
FEATURES INCLUDE :-



FORMATTED OUTPUT
 PROGRAMMABLE TAB FIELDS
 BI DIRECTIONAL SCROLLING
 INPUT '.,:.' TO STRINGS
 DISK STORE/RECALL COMMANDS GIVING -
 3 FOLD SPEED INCREASE
 AND UPTO 3 FOLD SPACE REDUCTION
 DIRECT ACCESS TO DISK SECTORS
 MID\$=STRING
 'HELP' FOR DEBUGGING
 'SCREEN' OR 'PAGE' OUTPUT
 THE MINI ASSMBLER
 USES ALL 68K OF RAM



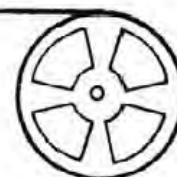
SEE MICROSOURCE FOR DETAILS
 IF YOU CAN RUN PASCAL YOU CAN RUN BUSINESS BASIC



kay-jay studios

135 COLES GREEN ROAD LONDON NW2 7HN

TEL: 01 460 6993



ANNOUNCING A NEW PRODUCT RANGE :-

=====

PLUGIT

=====

A range of interconnecting cables for the APPLE & its peripherals.
 The first in this range is an extension for the GAME I/O PORT; a DIL
 with 24 inches of ribbon cable terminating with a 15 pin D socket.

introductory price £19.95 inc vat p&p

Also available a DIL plug with 24 inches ribbon cable terminating with
 a zero force insertion socket; and RS 232 extension cables.

Telephone for latest information or S.A.E. for full range and prices.

TC JOYSTICK £35.00 inc VAT & p&p.

Reprinted from 'Creative Computing' magazine,

**Helmar Herman**

Many Apple owners have waited with great anticipation for an Applesoft Basic compiler. Now, within a span of a few months, four companies have released Applesoft compilers.

Why this interest in Applesoft compilers? And what is an Applesoft compiler anyway?

To understand what a compiler is, and what it can do for you, you must first understand how your Applesoft Basic works.

Interpreter

Applesoft is an interpretive language. Each time a Basic statement is to be executed by Applesoft, Applesoft must re-examine the statement to determine what to do. (Please refer to the sample program in Listing 1.)

```
10 INPUT 'ENTER A NUMBER';NUM
20 PRINT SQR(NUM)
30 GOTO 10
```

Listing 1. Sample Applesoft program.

Every time the program logic goes to a new line, Applesoft must examine the statement. For example, in line 10, Applesoft must determine that it is an INPUT request, that ENTER A NUMBER is to be displayed prior to prompting the operator, and that the response is to be placed in NUM.

When a number is entered, Applesoft will then examine statement 20 and perform the specified action (PRINT). Applesoft will then examine the next statement, 30, and perform the GOTO request.

Whenever a new line is encountered, Applesoft must re-examine the line as though it has never seen it before. It is because of this constant re-examination of statements that interpretive languages are slow.

Another problem with interpretive languages is that whenever program control is to transfer to a new statement (statement

30), Applesoft must spend time looking for the statement to which transfer is to be made. In a large program, this can take a considerable amount of time.

Compiler

A compiler transforms the Basic program into a machine language program. Each statement is examined by the compiler and is converted (compiled) into machine language instructions that perform the specified action.

When the program is run, the compiled machine language program is executed directly by the hardware, without a software interpreter having to examine each line.

When program control is to be transferred to a new statement, the address of the new statement is compiled into the machine language program, thus eliminating the need to search for the statement to which transfer is to be made.

Advantages and Disadvantages

There are various trade-offs with interpreters and compilers. See Figure 1 for a list.

In general, interpreters are much better when developing programs because of the ease and speed of program modification

and debugging. Compilers are better when the program is ready for production work because of the speed advantage.

Four Systems Examined

The four systems being examined are Expediter II from On-line Systems, TASC from Microsoft Inc., Applesoft Compiler from Hayden Publications, and Speed Star from Southwestern Data Systems.

In examining these systems, important items to look for are:

1. *Accuracy.* The compiled programs should run *exactly* as the interpreted programs do.

2. *Compatibility.* All functions of the Applesoft interpreter should be supported.

3. *Program Optimization.* The faster and smaller the compiled program, the better.

4. *Communication.* Compiled programs should be able to pass information to and from other compiled programs.

5. *Program Size.* Compiled programs will normally be larger than the interpreted version. If the compiler generates excessively large programs, its usefulness may be limited.

Item	Interpreter	Compiler
Ease of program development	Here the interpreter is far superior. A change to a program can be made and the results tested in just a few seconds.	Whenever a change is made to a program, the program must be re-compiled before testing can proceed.
Speed	Interpreted programs run relatively slowly.	Compiled programs run relatively fast.
Size	In general, interpreted programs are smaller.	Compiled programs are generally two to three times larger than interpreted programs.
Remarks	Remarks are to be avoided because they make the program larger and slower. On the other hand, programs without remarks are very difficult to modify.	Since remarks are removed during the compilation process, they may be used as needed.

Compilers, continued...

Slightly less important items of interest are:

1. *Speed of Compilation.* Since (in theory) compilation is performed only after the program has been debugged, the program won't be compiled very often. A fast compile is nice, but not a critical factor.

2. *Ease of Use.* The compiler should be

easy to use and forgiving in nature. It should be fairly safe to assume, however, that most persons using an Applesoft compiler must be familiar with Applesoft and know how to program.

Also of interest to software authors who may want to market compiled programs is the policy of the compiler company toward distribution of compiled code.

Expected Problems

Programs that call other programs that have been compiled will have to be changed to do a BRUN XXXXXXXX (or BLOAD XXXXXXXX:CALL YYYY) instead of a RUN XXXXXXXX.

Programs that depend on timing loops will have to be modified. For example, suppose a program uses the following statement to pause for a few seconds: 10 FOR X=1 TO 1000:NEXT X. With an interpreted version, this may take 2-3 seconds. With a compiled version, however, it may take only a fraction of a second.

Real-time games will have to be modified to adjust for the speed increase. How would you like it if all of a sudden, the balls in your favorite game started whizzing by at five times the speed you're used to?

```

5 REM TEST A
10 REM STRING MANIPULATION TEST AND TRY TO CAUSE APPLESOFT ER
20 HOME : INPUT "HIT RETURN TO BEGIN TEST";X$
100 REM FILL THE STRINGS
105 TEST$ = "A"
110 X$ = "X";Y$ = "Y";Z$ = "Z"
140 FOR COUNT = 0 TO 250
150 X$ = X$ + "X"
160 Y$ = Y$ + "Y"
170 Z$ = Z$ + "Z"
175 GOSUB 2000
180 NEXT COUNT
200 REM FILL ARRAYS WITH THE STRINGS
201 VTAB 12: HTAB 15: PRINT " ";
205 TEST$ = "B"
210 DIM X$(20),Y$(20),Z$(20)
220 FOR COUNT = 0 TO 19
230 X$(COUNT) = X$
240 Y$(COUNT) = Y$
250 Z$(COUNT) = Z$
255 GOSUB 2000
260 NEXT COUNT
300 REM STRING DOWN THE CHARACTERS
301 VTAB 12: HTAB 15: PRINT " ";
305 TEST$ = "C"
307 FOR COUNT = 1 TO LEN (X$) - 1
310 X$ = LEFT$(X$, LEN (X$) - 1)
320 Y$ = LEFT$(Y$, LEN (Y$) - 1)
330 Z$ = LEFT$(Z$, LEN (Z$) - 1)
335 GOSUB 2000
340 NEXT COUNT
1000 PRINT "TEST COMPLETE"
1010 END
2000 REM ROUTINE TO DISPLAY CURRENT TEST AND COUNT
2010 VTAB 12: HTAB 15
2020 PRINT TEST$,COUNT;
2030 RETURN

```

Listing 2. String manipulation test.

```

5 REM TEST C
10 REM TEST OF A BUBBLE SORT
20 HOME : INPUT "HIT RETURN TO BEGIN TEST";X$
30 DIM A(100)
100 REM FILL THE ARRAY
105 TEST$ = "A"
110 FOR COUNT = 1 TO 100
120 A(COUNT) = 101 - COUNT
125 GOSUB 2000
130 NEXT COUNT
200 REM SORT THE ARRAY
201 VTAB 12: HTAB 15: PRINT " ";
205 TEST$ = "B"
210 FOR X = 99 TO 1 STEP - 1
215 COUNT = X: GOSUB 2000
220 FOR Y = 1 TO X
230 IF A(Y) < A(Y + 1) THEN Z = A(Y)
240 A(Y) = A(Y + 1)
250 A(Y + 1) = Z
260 NEXT Y
270 NEXT X
280 NEXT COUNT
1000 PRINT "TEST COMPLETE"
1010 END
2000 REM ROUTINE TO DISPLAY CURRENT TEST AND COUNT
2010 VTAB 12: HTAB 15
2020 PRINT TEST$,COUNT;
2030 RETURN

```

Listing 4. Bubble sort test.

```

5 REM TEST D
10 REM SPEED TEST FOR HEAVY DISK I/O PROGRAM
20 HOME : INPUT "HIT RETURN TO BEGIN TEST";A$
30 D$ = CHR$(4)
40 TEST$ = "A"
100 REM OUTPUT THE FILE
110 PRINT D$;"OPEN TESTFILE.D1"
120 FOR COUNT = 1 TO 100
130 PRINT D$;"WRITE TESTFILE"
140 PRINT "THIS IS THE SAMPLE RECORD,";COUNT
150 GOSUB 2000
160 NEXT COUNT
170 PRINT D$;"CLOSE TESTFILE"
200 REM READ THE FILE BACK
210 TEST$ = "B"
220 PRINT D$;"OPEN TESTFILE"
230 FOR COUNT = 100 TO 1 STEP - 1
240 PRINT D$;"READ TESTFILE"
250 INPUT Z$,COUNT
260 GOSUB 2000
270 NEXT COUNT
280 PRINT D$;"CLOSE TESTFILE"
1000 PRINT "TEST COMPLETE"
1010 END
2000 REM ROUTINE TO DISPLAY CURRENT TEST AND COUNT
2005 PRINT D$
2007 VTAB 12: HTAB 17: PRINT " ";
2010 VTAB 12: HTAB 15
2020 PRINT TEST$,COUNT
2030 RETURN

```

Listing 5. Speed test for heavy disk I/O program.

```

5 REM TEST B
10 REM PROGRAM SIZE TEST
100 A=1
1010 END

```

Listing 3A. Program size test.

```

5 REM TEST B
10 REM PROGRAM SIZE TEST
100 A=1
110 B=2
1010 END

```

Listing 3B. Program size test.

Compilers, continued...

The Tests

Test A (Listing 2) is a string manipulation test. Being tested is the speed of the compiled version and its vulnerability to Applesoft frees. Since string manipulation requires a fair amount of work by the computer, I would expect a significant time improvement in this test. An Applesoft

free, by the way, is a problem that arises in large programs that do a great deal of string manipulation. As the area reserved for strings fills up, Applesoft periodically condenses the strings (also called garbage collection) to free up room at the end for more strings.

Test B (Listings 3A and 3B) is a test to see how large a single statement program

is and how much it grows by adding one simple statement. Listing 3A will show you how much overhead each program will have. To this overflow you then add the amount of storage taken by each compiled program statement. The lower the overhead, the better.

Test C (Listing 4) is a general speed test. It uses a bubble sort (one of the slowest) to test the speed of a logic-bound program.

Test D (Listing 5) tests the speed of a heavy disk I/O program. I expect that compiled programs will run at about the same speed as interpretive programs.

See Figure 2 for the results of the tests.

All the compilers tested shared the following features:

1. The compilers produce machine language object code that can be loaded and run with the BRUN command.
2. The compiled program can be placed anywhere in memory. Once compiled for a particular place, it can only run there.
3. Figure 3 shows what commands are not supported by which compilers.
4. Special compiler directive commands are supported via REM statements.
5. Compiled programs cannot be interrupted with Ctl-C.
6. They support local or global variables. Global variables allow you to pass information from one compiled program to another (but not from an un-compiled program to a compiled one).
7. Once started, the compilers can only be interrupted with the Reset key.

Expediter II

Expediter II comes with two disks. One for DOS 3.2, and the other for DOS 3.3. The disks are copy protected.

Rather than just producing a machine language program which is then BRUN, the compiler produces a one statement Applesoft program. The one statement is usually 1 CALL 4352. The machine language portion of the program is attached to this single line Basic program. You can thus SAVE, LOAD, and RUN the program exactly as you would any other Basic program.

If you *must* have a BLOADable version of the compiled code, there are instructions on how to accomplish this.

You can also leave "holes" in your compiled code to provide room for such things as the hi-res areas.

One potentially difficult problem is in the method used for string manipulation. Applesoft treats all strings as variable length strings. Thus a 5-byte string and a 60-byte string would take 65 bytes (plus overhead). Expediter treats all strings as fixed length strings. Before compilation you must specify how long the strings are to be. All strings will then be that length. Thus a 5-byte string would still occupy 60 bytes (or whatever string length was specified).

TEST	LENGTH	SPEED		
		(C1)	(C2)	
A(I)	717	23	••	Listing 2.
A(O)	3465	30!	33	
A(M)	4797	13	98	
A(H)	2500	(*1)	14	
A(S)	3222	17	1	
B(I)	38	••	••	Listing 3A.
B(O)	2362	••	11	
B(M)	4028	••	62	
B(H)	376	••	10	
B(S)	2101	••	1	
B(I)	46	••	••	Listing 3B.
B(O)	2384	••	11	
B(M)	4048	••	62	
B(H)	395	••	10	
B(S)	2124	••	1	
C(I)	464	124	••	Listing 4.
C(O)	2964	27	23	
C(M)	5049	40	94	
C(H)	2622	24	15	
C(S)	3070	53	1	
D(I)	594	36	••	Listing 5.
D(O)	3275	32	27	
D(M)	4642	30	94	
D(H)	1937	30	13	
D(S)	2730	30	1	

The top numbers (I) are the interpreter figures;
(O) are the On-Line compile figures;
(M) are the Microsoft compiler figures;
(H) are the Hayden compiler figures;
(S) are the Southwestern compiler figures.

Note—the compiled lengths for each compiler are calculated differently. Length comparisons may not be completely accurate.

(O) compiled length excludes variables.
(M) compiled length includes variables.
(H) compiled length includes non-string variables.
(S) compiled length includes variables.

(C1) is the program run speed.
(C2) is compilation speed (how long does it take the compile).

*When Listing 2 was compiled and run on the Hayden compiler, an error occurred after the 147th loop of test A. The error was OUT OF MEMORY ERROR IN MODULE \$0803. In accordance with a suggestion from the manual, I added the following line the program: 145 X=FRE(0). The program then ran in 15 seconds.

Figure 2. Test results.

Compilers, continued...

The good part of this is that there is never any garbage collection. Also, in theory, string operations should be faster. In practice, however, they appear to be slower.

The bad part is that if you have many strings, they must all be as long as the longest one, which may cause a storage problem.

Worse, is that you are not notified if you exceed a string length. The program just keeps on running, wiping out who knows what variables until eventually something vital is destroyed and the program fails.

There is an unusual restriction with this system. All arrays must be defined in the program *physically* ahead of the first use rather than *logically* ahead of the first use. Thus the following program is invalid:

```
10 GOSUB 100
20 A(20)=5
30 END
100 DIM A(20)
110 RETURN
```

The DIM statement at 100 must physically precede the first use at statement 20 even though statement 100 will be executed before statement 20.

Because of this restriction, and because of the common practice of placing DIM statements at the end of programs (for speed), inexperienced users may have trouble with Expediter when compiling off-the-shelf programs.

As far as speed and length of programs, the Expediter sits comfortably between the extremes produced by the tests, though it created the slowest code for the string program. On the other hand it has the fewest unsupported Basic statements of any of the compilers tested.

An annual charge of five times the list price is charged for distribution of compiled code.

On-line Systems, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. \$99.

TASC (The AppleSoft Compiler)

The version of TASC that I tested was a pre-release Beta test version, and all test results should be viewed in that light.

TASC is distributed on a DOS 3.2 disk. It can be muffed to produce a DOS 3.3 version. The disk I have is not copy protected, and Microsoft has indicated that they have not yet decided whether or not they will copy protect the final production version.

TASC was written in Basic and then used to compile itself—a very intriguing concept.

The output from TASC is a relatively small BLOADable file. The size of the object program is deceptive, however, because in order to run the program, you must first BLOAD the file RUNTIME. RUNTIME contains execution time sub-

Unsupported Statement	On-line	Microsoft	Hayden	Southwestern
CONT	U	U	S	U
DEF FN	S	*	*	S
DEL	U	U	U	U
HIMEM	U	S	S	S
IF X\$ THEN	S	***	S	S
LIST	U	U	U	U
LOAD	S	U	U	U
LOMEM	U	U	S	S
NEXT	**	S	S	S
NOTRACE	S	U	U	S
RECALL	U	U	U	U
RESUME	U	S	S	U
SAVE	S	U	U	U
SHLOAD	S	U	U	S
STORE	U	U	U	U
TRACE	S	U	U	S
&	S	U	S	****

U = Unsupported
S = Supported

*DEF FN may be defined, but not re-defined.

**A FOR may have one and only one corresponding NEXT. The following program would be invalid:
 10 FOR X=1 TO 10
 20 IF X=5 THEN NEXT X:GOTO 40
 30 PRINT X:NEXT X
 40 END

***Strings may not be used as a logical argument. Statements such as IF A\$+B\$ THEN 100 are illegal. Statements such as IF A+B THEN 100 are legal.

****Makes the compiler crash.

Figure 3. Unsupported Applesoft statements.

outines which are called by your compiled program. It is approximately 4K long.

It is a slight inconvenience to have to BLOAD RUNTIME every time you want to run a compiled program. On the other hand, the amount of disk space required for each program is reduced.

For frequently used programs, you can simply write an Applesoft program that BLOADs the runtime package and then BRUNs your compiled program.

Unlike the other compilers, TASC can be gracefully interrupted while compiling.

Whenever the program pauses for user information, such as program name, you can enter DOS command by prefixing the response with a Ctl-D.

TASC fared worst in compilation speed, but did reasonably well in execution time. While the runtime routines take a fair chunk of space, the compactness of the compiled code could make up for this in long programs. TASC had more unsupported statements than any of the other compilers.

Microsoft, Inc., 10800 NE Eighth Suite 819, Bellevue, WA 98004. (206) 455-8080. \$150.

Hayden Applesoft Compiler

The Hayden compiler is currently available only in 3.2.1 format. Hayden indicates that a 3.3 version is coming and will be shipped free to users who have returned their warranty cards.

Although the disk is not copy protected, the system is shipped with a special "protection device" that must be installed in the game I/O socket.

Hayden was not specific about charges for re-distribution of compiled code. The impression I got was that each request will be handled on an individual basis.

Two extra programs are supplied with the system. The first is the only full color, low-res, single disk copy program I've ever seen. It's quite entertaining. Second is a program to "de-muffin" programs from DOS 3.3 to DOS 3.2. I expect that the second program will disappear on DOS 3.3 versions.

This compiler was the second fastest in compilation and registered favorable execution speeds. On the other hand, each time you want to re-compile a program, you must re-boot! If you are already booted, there is no way to invoke the compiler without booting again. Also, when the compile is done, you must hit the Reset key to exit—a rather strange idea.

The ability to pass data from compiled program to compiled program is limited. All numeric variables must be referenced in the respective programs in exactly the same order. If string variables are shared, then `FRE()` statements must be inserted into the programs at strategic points.

Like the Microsoft compiler, this one also generates a set of runtime subroutines. These subroutines can either be included in the program or be declared `EXTERNAL`. If they are external, then they must be `BLOADED` before execution.

Speed Star

The version of Speed Star that I tested was also a pre-released test version. All test results should be viewed in that light.

Like the Hayden compiler, this one is protected by a device inserted into the game socket. According to Southwestern, the final version will allow you to plug your game paddles into the protection device, thus allowing both to be attached at the same time.

This compiler is lightning fast. The test compiles were done almost instantaneously. Also, repeat compiles are quick because you don't have to reload the compiler each time. However, comparative execution speed varied considerably from test to test. The compiler locates itself at location `X'7200'` and is invoked with the "8" key.

The address at which Speed Star runs is `HIMEM` for a 32K system. Unfortunately, it still loads there on a 48K system. The extra 16K does not appear to be useable during the compile process, thus limiting the size of your program.

One nice feature is the ability to include `Ctl-C` checking logic in the object program automatically. Also, you can have the system check subscribing ranges. These checks, of course, do not come free. They result in a decrease in storage and a slight speed degradation.

There are many times when a significant speed increase can be accomplished just by having one or two Applesoft sub-routines converted to machine language and leaving the rest of the program interpretive. This was the only compiler that allowed interpreter programs to call and pass variables to and receive variables from compiled programs. Although I didn't actually try this feature, it seemed fairly simple, based on the documentation.

Southwestern Data Systems, P.O. Box 582-S, Santec, CA 920071 (714) 562-3670. \$85 (introductory price).

Conclusions

I ran one other test on all four of the compilers in which I tried to compile the menu program from The CCA Data Management System. The program works perfectly in interpretive mode. After compiling, I got the following results:

Microsoft: It worked for a while. I was able to define the system configuration, but then it crashed into the monitor.

Online: The screen went blank and then nothing.

Hayden: The menu was displayed, but whenever I tried to enter a number for a valid function, the system beeped and rejected it. Then when I entered an invalid function, the program got an Applesoft error.

Southwestern: The configuration portion ran, but instead of displaying the menu, the program terminated.

Just as an educated guess, I suspect that the compilers are having trouble with the `ONERR` routines.

I recompiled with the Microsoft compiler and specified inclusion of the `RESUME` logic. The program worked much better and operated correctly with one minor exception.

At about this time, Microsoft sent me some fixes to their compiler, and wonder of wonders, the problem was solved. The menu now worked perfectly.

I was encouraged by this and tried compiling the `SORT` portion of the system. A dramatic success. The compiled program ran perfectly and was substantially faster than the interpretive program. I did have to modify the program to change the dynamically dimensioned arrays to statically dimensioned arrays.

The above experience illustrates my concluding statements and conclusions:

The larger and more complex a program is, the less likely it will compile successfully without modification (for example, dynamic arrays).

For one reason or another, off-the-shelf programs will probably not compile successfully without modification and effort.

Recommendation

Before I applied the patches to the Microsoft compiler, I was not ready to recommend any of the compilers.

After running the fixed program, however, the Microsoft compiler would be my recommendation. Although it is by far the slowest, the programs it compiles seem more likely to run successfully. It appears that once again, the tortoise has beaten the hares!

If you have programs that are too large to compile, you may want to consider the Southwestern compiler. It is the only one to support the compilation of subroutines with full data transfer capabilities. As was evidenced with the CCA DMS main menu, however, you may have to fiddle with your program to get it to work. □



**SPECIAL
OFFER
TO BASUG
MEMBERS**

**VALID UNTIL
31 / 1 / 82
ONLY**

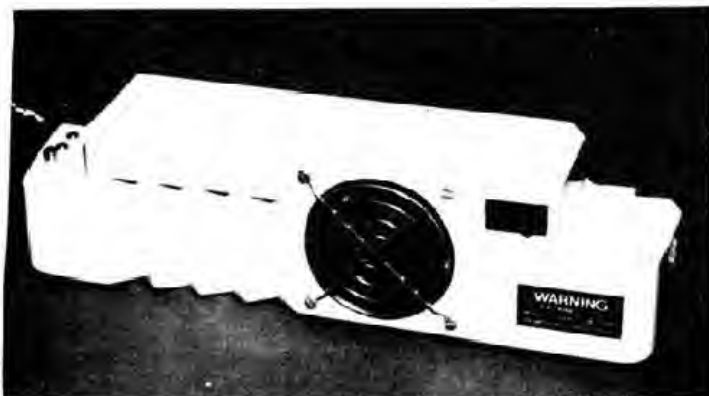
**APPLEFAN
BASIC UNIT
£49.50
+ VAT & P+P**

**SEND YOUR
CHEQUE
FOR £58.65 TO**

**BASUG,
P.O. BOX 174,
WATFORD,
WD2 6NF.**



THE **APPLEFAN** UK



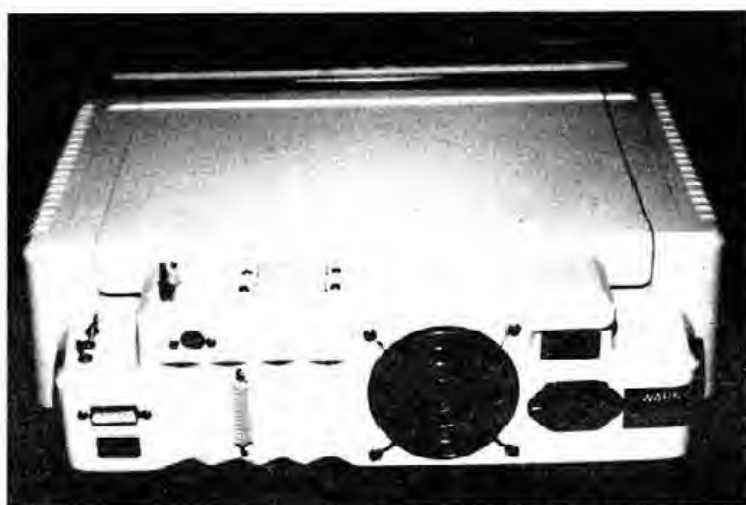
The **APPLEFAN** unit is an easy to fit accessory which reduces the risk of machine failures due to over heating when using certain interface cards.

The **APPLEFAN** also enables the mass of wires found at the back of most APPLE II systems to be tidied up. Making a neater and more reliable system.

No soldering or major changes are needed to fit the **APPLEFAN**.

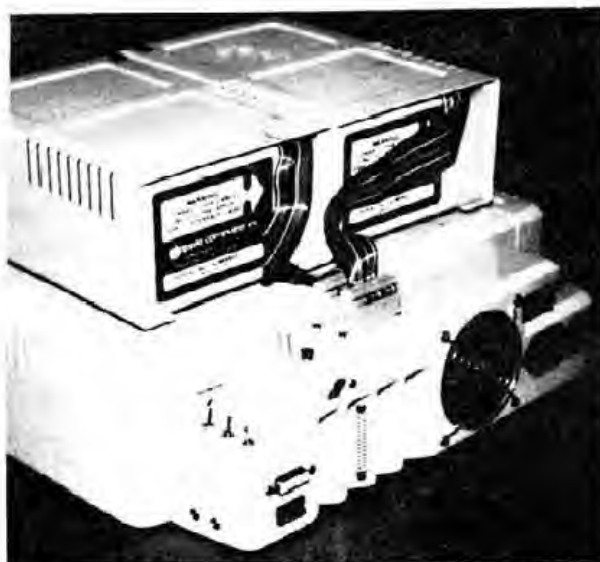
FEATURES:-

- Low Noise High Quality Fan. Only 30db. Quieter than most Disc Drives.
- Cooling Keeps components at a Low **SAFE TEMPERATURE** even with cards in every slot.
- Appearance Reduces mass of wiring at Rear of APPLE II. Matches colour & style of APPLE.
- Easy to Fit Can be fitted in 10 minutes. Plugs in - **NO SOLDERING** - Rigid fixing to APPLE II.
- Additional Connector Kits Available.



CONNECTOR KITS:-

Connector & System kits are available which can be fitted to the **APPLEFAN** which allow devices such as Games Paddles, Graphics Tablet, Monitor Power Supply, Disc Drives etc. to be plugged and unplugged from the Computer without having to remove the Top Cover. Space is available on the **APPLEFAN** to fit connectors for:- up to 6 Disc Drives, 2 Printers, Monitor Power Supply, Switched Monitor Outputs, Games I/O, Cassette Interface, and two 25 way outputs for additional peripheral devices.



Basic APPLEFAN unit (includes Mains Input, On/Off Switch, Monitor Output, Fan, Fixing Screws, Templates and Instructions)	£62.00
Disc Drive Connector Set	£16.00/pair
Cassette Connector Set	£4.25
Printer Connector Set	£16.00/pair
(25 way 'D')	
Monitor A/B Box Set	£10.50/per set
(switch between 2 inputs - 1 output)	
Games Connector Set	£9.80/per set
Monitor Power Supply Kit	£7.00/per set
25 way 'D' Connectors	£7.90 per socket
(IDC for .05 pitch flat cable)	£7.95 per plug
Retail Prices Excluding VAT Dealer Discounts Available	

HI TECH PRODUCTS

Apple is a Registered TM of Apple Computers, Inc.
21a Station Road, Knebworth, Hertfordshire, England.
Telephone: Stevenage (0438) 812137



EPSON

MX80 Series Impact Printers

NEW REDUCED
PRICES

BASUG
members
**Special
Rates**



Great Printer Offers

12" GREEN
DISPLAY

ex
stock

BMC
£120 + VAT!

80chr x 24 anti-glare
18 MHz band great for Hi Res!

Brightness
Contrast
Power

LED Power
Indicator

MONITORS

NEW
BMC
RGB

COLOUR!!

SPECIAL OFFER £230 + VAT

VIDEX

80 COL CARD

£165 + VAT

floppy disks

SPECIAL OFFERS!

WE ALSO SUPPLY

COMMODORE PET

&

SUPERBRAIN

- stand alone system

Demonstrations Arranged

CALL US
FOR
DETAILS



£998 + VAT

**£150 OFF
KSR**

correctable
ribbon
+ many
features

RD
£100 OFF

SCRIPTA

A Word Processor for the real world

zardax™

THE BEST WORD PROCESSOR

£150.00

FREE DEMONSTRATION
Local Support and Service

+ VAT

MASS MICROS

MICROCOMPUTER SALES, SERVICE,
TRAINING AND CONSULTANCY

Wellson House, Brownfields, Welwyn Garden City, Herts. AL7 1AR
Telephone W.G.C. (07073) — 24363 Telex 298641

Reader's letters

Reigate

(If your letter is here and you expected a reply personally, DID YOU SEND A SAE? If you didn't I treated it as a letter for HARDCORE. If you have a problem that requires immediate answer, it is easier if you can go to a local group meeting. If there isn't one near you, start one. Or, if there is a contact near you, badger them to get something going and let us know what is going on. Meanwhile, keep the information flowing in, you are doing well, John Sharp)

Rickmansworth

Dear Sirs,

Just over two weeks ago I sent a cheque for several disks. As yet I have not received them. I am concerned, as the last time I ordered a disk I waited well over a month before it arrived.

If the demand for software from the group is high could not the group employ someone on a part time basis to deal with demands for software.....?

Nigel M Cook

(As you can see from the agenda of the SGM we are looking to take on some paid help. But at the moment we are struggling at times. The post varies, and can be up to 100 letters a week. Unfortunately, the week after a magazine goes out it tends to be inflated with orders and as we are voluntary something has to give. Occasionally, orders do go astray, as was the case with your first one. Considering the numbers of disks etc we do send out we do not do too badly. Then there are problems because we are spread out. This is especially true of the disk copiers so if a particular disk runs out there will be up to a week before fresh supplies arrive. Please be patient, we are doing our best. Please wait a month before complaining, but do complain if you feel it isn't good enough, and if you live close to Watford, we can always do with some help.

Sheffield

Dear John,

... As you know Len Gould and I have just about got a small group off the ground up here ... we would both be interested in a resurrected Pascal course - so keep us informed.

Finally, just a small goody which you might like to publish if not already in widespread use!

POKE 1013,76) Built into "Hello" for instance

POKE 1014,165) will give listing by typing

POKE 1015,214) "<Return>".

Cheers,

G. Reidford.

Mons, Belgium

Dear John,

I find that the Menu 2.1/48K program on the Derver DOM Jan 81 is an excellent menu program. However, to make it work correctly in DOS 3.3 you need to modify the routine for the free sector count as follows:

BLOAD MENU 2.1/48K

CALL-151

9323:96

3D0G

BSAVE MENU 2.1/48K, A36608,L1776

It can also be used as a "Hello" program if you modify DOS by 9E42:34

Sincerely yours,

Richard Sylvester

(Ed. Thanks for the tip. WE NEED MORE. A number of others have written in saying they are converting the utility disks to 3.3 DOS. We do not have any idea yet of how many members are still on DOS 3.2, but the disks as we get them are still 3.2. We would like time to do something ourselves, but.....if you are doing the conversion let us know, and send them in; you will get a credit for doing so.)

Dear Sir,

Thanks for the copies of Hard Core and the introductory disk. I liked the articles about Peeking and Pokeing. They gave me some insight into what these terms mean. Having only a very basic (sic) knowledge of Applesoft I thought these were a method of cheating employed by people who know a lot more about it than me. The Apple manuals do not seem to be very comprehensive on the subject although I suppose someone who knows what they mean can find the information in the ROM listings. I do have a problem with my machine that I hope some will be able to help me with.

On the DOS 3.3 master disk there is a program called INTBASIC. When I BRUN INTBASIC the screen goes into HGR mode covered in small grey dots and the only way I can recover is to hit the RESET button. I would be very pleased if someone could explain this problem as I think I may have a corrupt disk or possibly something wrong with the Apple.

Yours faithfully,

Martin Stoggeil

(Ed. I hope that your questions are answered with this month's BEGINNERS PAGE. You are not the first to come back with this query. I hope the tape users understand what is going on as well, particularly when they see the catalogs printed out in the software library....John Sharp)

Maldon, Essex

Dear Mr Williams,

I'm trying to compile an anthology of biology program listings for publication, for educational purposes and would be grateful to be put in touch with anybody who might be able to help...

Dr P J Baron

(Ed. any contributions? Please send them via our box number)

Bottesford

Dear Sir,

Apple Machine Language a Reply Tut, tut, Ian Trackman - were you never a rank amateur, thirsting for knowledge of the esoteric ways of 6502 language? I suspect you learned to be a professional programmer in the same way that I entered my profession, by a course of full-time study including degrees and that sort of thing. I share your feelings that we should aspire to high standards and I agree that the book 'Apple Machine Language' by Don and Kurt Inman is not of a high standard if judged by its typographical errors and its inability to turn a novice into an expert.

I do, however, think that it is a worthwhile introduction to Apple machine code for someone who is acquainted with BASIC but knows nothing of peeks, pokes, assemblers, etc. Some of the bugs in the text are awful and should never have been allowed in a book of this type, but even with my limited knowledge I have readily spotted them (I think). I would not call this a deep book in the sense that it provides little information on the architecture of the machine, and it is of little use for reference as it has nothing of lasting value in the appendices, such as detailed lists of intrinsic subroutines and how to use them. The authors' comments that having read the book you will be able to dig these things out for yourself is, I believe, a little naive, at least considering my own abilities. The price by the way is about £11.00 which makes it an expensive paperback.

Finally in answer to Ian Trackman I would say that those of us who have become 'experts' in our own fields sometimes find it difficult to climb down from that position and to try to see the world through the eyes of those not so skilled. This book does offer a worthwhile introduction, but I must ask: having found out a little about machine language, what book would he recommend to pursue the subject further?

Kind regards,

David Durling

PS I am writing this using the Correspondent; for any future letters

can I send a binary file or a text file for you to use with Applewriter, and are there any special constraints?

(Ed. On the subject of Applewriter files, we would prefer you to send us an Applewriter disk, which we will return. Better still, as you are using a Centronics 737, why not make life easier for us by sending us a fill justified, condensed character style print out exactly as you see it here. It is not our policy to edit your letters, if we can avoid it, so why make us re-type? One thing: please use a fresh ribbon.

There has been no time to get a reply from Ian Trackman. We have, however, had a reply from Prentice Hall, who distribute the book in this country to the effect that a NEW REVISED EDITION has been published, which hopefully has had all the mistakes checked. Make sure you get the corrected edition.

Rayleigh, Essex

Dear BASUG,

...The secondhand Apple disk drive I have bought had about 30 disks with it, but no instructions with any of the programs. Can you help? In particular, the problem programs are: Forth, Pilot, Applewriter, ITT 48K Assembler.

The shop where I bought the language card were adamant that the integer card could only be used in Slot 0, just where the Language card had to go too. Sell it off, they said, can't be used! I don't know if any other members have had this happen to them. Well, after a few evenings fiddling with DOS I found that for DOS 3.3 on 48K ITT the following POKES will tell DOS that the Integer Card is there instead, thus avoiding the necessity of loading integer into the card at all.

POKE 42424,160

POKE 42432,161

(N.B. 160=128+N*16); 161=129+N*16 where N is the slot number.)

After these two pokes any disks INITed will contain the fixed DOS but not if you run MASTER CREATE. To do this I recommend reading "Beneath Apple DOS".

... If there are any handicapped Apple users in my area that I can offer any help or assistance to, please let me know.

Mike Norfolk

Thornton Heath

Dear Someone,

May I add to the general discussion on two points: 1) Magazine name: I consider Hard Core to be a brilliant choice - don't change it. 2) Bit-copiers. I confess that I am thinking of buying one so I can make back-ups, especially of SDC III. Personally I would not be involved in any sort of piracy, as even the simple programs that I write have shown me just how much effort and time goes into the average commercial software. I would suggest that the simple solution would be for all packages to be supplied with at least one back up disk. This would at least provide the user with continuity while arranging for replacement of a damaged / worn disk. While this would not entirely stop piracy it should reduce the incidence of it. There will always be pirates. The addition of a second disk would not significantly increase the cost of a package and in fact, judging from some prices could be supplied at no additional cost.

May I thank you for what I consider to be an excellent and absorbing magazine.

Your faithfully,

John F Tomlinson

Wilmington, Kent

Dear Sir,

Many thanks for the disk and issues of Hard Core; the first issue solved several problems I had encountered, e.g., how to draw on one page of Hi-res while displaying the other.

I read with interest of "The Go-Between" for use with Apple Writer and the Centronics 737 and wondered if anything similar existed for use with the Magic Window word processor and the same printer. I have tried writing a Basic program using the soft-copy option of Magic Window to provide justified print, but with little success. Can anyone help as there is little use in having proportional print if it cannot be justified?

By the way, this was written on Apple Spiel, a word processor written in BASIC for the Apple and the 737 and I find it ideal for short letters.

Roland Horwood

'Ed. Hardcore has been taking a first look at the Magic Window as well and our findings are printed elsewhere in this edition. Has anyone solved Roland's problem? If so let us all know. Neil Lomas who wrote APPLE SPIEL has given a copy to the library. We just haven't got around to distributing it yet.)

London, SW16 4UN

Dear David,

In Hardcore number 5, John Sharp posed the question, "what use is there for CALL -958 (clear to end of page)?"

I can assure him that this CALL is one which I frequently use. Indeed it should normally be the CALL of choice whenever one wishes to erase an erroneous response to a program's request for something to be typed in.

The programmer cannot usually tell whether the user will type in more than will fit on the one line. If the entry overflows the line then CALL -868 will not clear all of it. CALL -868 should only be used if a) there is material below the affected line which cannot be otherwise protected but must not be erased and b) the erroneous entry cannot spill over onto the next line.

The example given in the article, viz,

```
30 CALL -868: INPUT "HOW MANY ANIMALS
DO YOU HAVE ALL TOGETHER ";A$
```

will not work, since the prompt string is 41 characters long and hence wraps onto the second line, which will not be cleared! This is the very place for a CALL - 958!

Your own comment on page 10 of Hardcore 5, in respect of the 'protection' built into the menu of Mr McCarthy's BASIC Tutorial disk misses totally the point of this 'protection'. The use of control characters in the menu makes no difference whatsoever to the direct copying of the disk using any normal copy utility. Both COPYA and FID (using the wildcard option) will cope perfectly well with unusual CATALOG entries. I agree that it is more

difficult to MUFFIN the programs across when this type of protection is used, but this must be offset against the advantages from using this type of catalog entry.

The purpose of the hidden letters and control characters is the very laudable one, for such a disk, of making it student proof. Mr McCarthy knows that if he does not protect the CATALOG entry from easy access, someone among his students will change the name of a file or even delete it from the disk! For a similar reason, the program disables both CTRL-C and the RESET key. If only more educational software were written with such safeguards built in!

If any member wants the Tutorial in D.O.S. 3.3 form and cannot otherwise obtain it, I have a copy direct from Dublin which I can duplicate at the cost of disk plus postage.

Yours faithfully,
Vernon Quaintance

East Ham

(Ed. Thanks for pointing out the error, Vernon; when you are writing, with Apple Writer, it is a bit of a problem popping back and forth to test what you have written, if time is short, as it always seems to be.

On the subject of the BASIC TUTORIAL, although I agree with you on the fact that the disk could be corrupted, the FID or MUFFIN programs are not as easy to implement as you suggest. Why? Because the disk needs to be initialised with the HELLO program as HELLOhhhh where h is a CTRL-H. If you do not do this the turnkey will not operate. But this, as you say is a minor point compared with the possible consequences of not doing it. You seem to be speaking with the voice of experience! Have you lost anything valuable by any chance?, John Sharp)

Livingston
W Lothian

Dear Sir,

Miscellaneous comments which may or may not be of interest.

1) I like Hard Core, but not the title - could it be called something else?

2) Tony van der Kuyl wants to coordinate an East of Scotland regional group. You ask "anything we can do?" Yes, there is, you could print his address! In fact, I should like to write to one or two of your correspondents, and would appreciate full addresses for all.

3) I have read Doubledos with interest. Is there a type of DOS which will allow a disk to be run on both 3.2 and 3.3 controller cards?

4) I have recently been using "Apple Machine Language" penned by Ian Trackman. I found it a good and understandable introduction to machine language and assembly language processing. A major criticism is the lack of any reference to high resolution graphics. Also a table of ML instructions at the back has a column for page refs of first occurrence or explanation. Unfortunately these are all blank! My summary as a beginner would be, Excellent for a beginner.

Your good work is much appreciated, keep it up.

Best wishes,
Graham Dane

(Ed. Thanks for a thoughtful contribution to the debate, Graham. But we do publish the address of members; it is PO Box 174, Watford. In other words we have a rigid embargo on publishing members' addresses without permission, but will re-direct all communications. Meanwhile contact Tony Van Der Kuyl on 0181 871111. And Tony- let us know what is going on.

On the subject of the disk to be run on both 3.3 and 3.2 controller cards, there is a program called uniboot which allows you to boot a disk with both controller cards but leaves you in a 3.2 environment. How about someone writing up how it is done, and where it comes from. OSNIDOS- see Microsource Advert- will boot on either and will access both types of disks automatically.)

East Ham,

local rather than having to communicate with "Head Office".

London NW8

Dear John,

... Please take the following comments on the understanding that they are not criticism since I do appreciate the amount of work involved in trying to meet everybody's needs.

1) While each issue sees a further nine or ten discs available a frightening amount of info to wade through, the "Catalog" very often gives no indication of what the programs are or do, for example, on disc No 23 is a suite of programs called "Tough" (with various versions). It is only if one happens to have specific issues of NIBBLE that one would realise that this is a primitive word processing program. Other titles are quite mind boggling as to what they might do.

2) Several titles appear on more than one disc, sometimes with a different number of sectors, which suggests that they may or may not be the same program, or different versions of the same program.

3) Some (such as microLisp?) cannot be expected to be usable without access to documentation of some sort, either standard references or to magazine articles, etc. from which the particular program is sourced. There are few more frustrating exercises than loading a program only to be faced with a blank screen and a sequence of INVALID COMMANDS whatever key is pressed and with limited or no info on how to get the best from the program. It is not clear which programs require such access or how such documentation is to be obtained. To overcome these difficulties will obviously require a great deal of reviewing of existing library discs and documentation, and this workload will need to be spread around a bit. Perhaps this is something that local units of BASUG could undertake to review if allocated specific disks, or a type to suit their particular members' aptitudes or interests.

You might also consider using such groups as distribution points for copies of discs to members which could reduce the workload on yourself and, I feel, would produce a greater feedback in the form of adaptations and improvements to the library programs. Failing a local group as such, perhaps an individual could act as a local distributor and collator. Many members would be more forthcoming to somebody

Similarly they (local groups or individuals) might be happy to reproduce as a matter of course the major programs published in NIBBLE, Creative, Call Apple or whatever, if they could be sure that somebody else is not already doing so. I have myself spent some 10 hours (mis-(typing) the aforementioned TOUCH programs, plus another 10 or so finding and correcting my bugs and some of those in the printed edition. It still needs some work in improving input error traps, etc. so I am hoping that the library version contains some of these features in the "enhanced" versions. The number of bugs or weak features in some of the other library programs leaves me in some doubt, however.

I don't think that there is any way that the mass of programs being issued can be properly vetted and "tuned" without involving a fairly large number of people, preferably in an interactive sort of environment, hence the thought of using the smaller and more intimate grouping of a local unit for this purpose. It should also produce a few more contributed software items if members can "try them for size" with one another before plucking up the confidence to submit them to the library.

...Having proposed some solutions I now, of course, have to put my money where my mouth is and offer to do some of the typing, vetting and, possibly, modifying any of such programs that you would suggest or allocate suitable ones, or to go through existing library discs with a view to rationalising titles, operation instructions. This I would gladly do if time is not too critical a factor, and if the programs are not too complex for my tiny mind.

I would also like to help with the credit rating system for the software library.

Yours

Ted Lopley

(Ed. You have made some very valid points, Ted. I hope you have had a word with John Rogers by now and things are starting to move on this front. If there is anyone else willing to help, please write to John Rogers. The sooner the work is started the sooner the smaller will be the moving target.)

My company is a Licensed Dealer in Securities, and we have written a program for calculating and printing contract notes for trading in stocks and shares. We would like to contact other writers and users of Apple financial software (?forecasting gold, commodity prices, charting, plotting moving averages, portfolio management, etc) - with a view to discussing common interests and, possibly, to co-operate on joint marketing. We are proposing to direct-mail other Licensed Dealers about our own program; it would be nice to offer a range of programs likely to be of interest to the financial community. My company would be prepared to set up a software division to provide marketing facilities for specialist programs likely to appeal to City and investment institutions. Please write to me through BASUG.

Gerard Noel

SPECIAL OFFER TO BASUG MEMBERS

Turn your Apple II
into a PRESTEL
Terminal

MICROTANTEL
Prestel Adaptor
with Microcomputer
Interface

£160 + VAT

MICROVIEW
Software package
to interface the
Tantel Adaptor to
the Apple Computer
(includes demon-
stration program)

£20 + VAT

Send cheque for
£207
(£180 + £27 VAT)
to-

BRUCETEL,
22 GILLS HILL,
RADLETT, HERTS.

Book Reviews

STRUCTURED BASIC AND BEYOND

reviewed by Michael Tickle

I wish I'd had this book two years ago at the start of my struggle to learn Basic programming.

Wayne Amsbury writes on the way to program effectively clearly, concisely, and with obvious enjoyment and, by default, shows me all the bad habits I have got into. The Basic used is non-specific to any machine, but variations are discussed where appropriate. The first third of the book describes Basic with examples using flowcharts, listings, and more importantly Pseudo Code. The rest of the book uses mainly Pseudo Code to explain the more complicated algorithms and is none the worse for that. The Pseudo Code as defined, forces the programmer to think clearly about what the program is required to do, and at the same time imposes a logical skeletal structure on the Basic Coding.

Some of the chapter headings give an idea of the range of algorithms covered, e.g. Loops and Structures, Files, Strings and linked lists, Stacks and Queues, tree-like Structures. Each chapter has self review, problems to solve and projects before moving onto the next chapter. Answers to the self review segment are at the end of the book.

Structured Basic and Beyond is aimed at university students and lecturers, to help produce professional programmers. But it is invaluable for 'do-it-yourselfers' and is the best book on programming without a doubt. I must alter my existing programs, if only to make them more understandable. If, of course, I had the time.

1. INTIMATE INSTRUCTIONS IN INTEGER BASIC, 158 pages

2. APPLESOFT LANGUAGE. 254 pages.
by John Sharp

Both paperbacks by Brian D. Blackwood and George H. Blackwood and published by Howard Sams at £5.55 and £7.65 respectively.

We start our book reviews this month with two hot off the press!

A brief perusal suggests that these are excellent books both for the beginner and the programmer who basically knows his way around and is looking for ways of developing programming skills. The Integer version is written for APPLE II owners and is in many ways a condensed edition of the Applesoft version. It explains why Integer is different and limited and guides the reader through the basics of clearing the screen and using the PRINT statement down to complicated string handling, sorts etc.

The Applesoft version is a little less elementary in its approach, to begin with. It seems to assume that you have worked through the Applesoft Tutorial and it moves quite rapidly into formatting and variable usage. By the time you have reached the end of the book you will have worked hard and covered a wide range of topics.

Both books have a similar chapter structure, a list of objectives, a vocabulary to explain new words and phrases and a discussion section - the actual meat of the chapter. The Integer book also has exercises at the end of each chapter. The example programs as they are built up are explained well, and there are programs which will not work and an explanation why they won't. The stress is on program design, efficiency and flexibility, and flowcharting is not only explained but actually used. There are useful chapters on the use of lists, sorting, arrays, string handling but only a brief supplement on graphics, Hi and Lo-res. The chapters on using flags, stacks and pointers are a good preparation for machine code programming. The Integer book is not likely to sell too well in this country, since it is only useful if you have an APPLE II or wish to program in Integer and haven't got an Integer manual.

The Integer book contains an index, the Applesoft does not.

The Applesoft book is a natural successor to the APPLESOFT TUTORIAL, and well worth even the non-beginner taking a look at.

NEW from MICROSOURCE!

What's Where in the Apple?

An Atlas to the Apple Computer

By William F. Luebbert

Adjunct Professor of Engineering, Dartmouth College
The MOST DETAILED description to date of Apple II
firmware and hardware.

- The Alias and Guarantee of PEEKs, POKEs, and CALLs is in tabular form over 2000 memory locations.
- Information is presented numerically in the Alias column alphabetically in the Guarantee column.
- The names and locations of various Monitor, DOS, and BASIC and Apple II routines are listed in the table.
- Information is provided on their use.
- The easy-to-use format includes:
 - The address in hexadecimal (useful for assembly programming)
 - The address in signed decimal (useful for BASIC programming)
 - The common name of the address or routine (useful for assembly programming)
- The common name of the address or routine (useful for assembly programming)
- Information on the use and type of routine (USE)
- A description of the routine (CLEAR SCROLL WINDOW TO BLANKS SET CURSOR TO TOP LEFT CORNER)
- Related register information: (A - Y - REGS ALTERED)

This reference tool offers information every serious Apple user needs. BASIC and assembly language users alike will find the book helpful in understanding the Apple.

128 pages 8 1/2 x 11 inches
\$9.95

**1 Branch Road, Park Street,
St. Albans.
Tel: Park Street (0727) 72917**

APPLE PANIC

A Review by Vernon Quaintance

APPLE PANIC is a disk based high-resolution colour graphics game from Broderbund Software.

When the disk is booted, the opening title appears on the screen. There is a man in the bottom right-hand corner and an apple on the left.

Whilst the main program is still being loaded from disk, the apple crawls across to the man, runs up him and jumps on his head! A very nice piece of programming to whet one's appetite for the game itself. Once loaded fully, the disk can be removed from the drive.

The game area itself consists of the side view of five tiled floors, linked by ladders of various lengths. Several hostile little apples are distributed over these floors, and soon start chasing you.

The object of the game is to destroy the apples by digging holes in the brick floors for them to fall into. Having trapped an apple in a hole, one must then hit it through with one's spade in order to kill it. To make things harder, if one delays too long before knocking the apple down then it struggles free and continues the chase. Knocking an apple through a hole, or having it free itself also results in the hole being filled in.

The man's direction of movement is controlled using the "I", "J", "K" and "M" keys, which work exactly as expected. To dig a hole, one must press "A" whilst the man is moving; however he will only dig if his feet are in the correct position. Most of the children who helped me with this review thought getting the feet right was too hard. To knock an apple through a hole, or to fill up a hole, one must move the man to the edge of the hole and then press "S".

One can jump through holes without injury, this is a very useful way of avoiding an apple or of getting quickly to a lower level.

If one can dig holes exactly one above the other then extra points may be obtained by knocking the apples down through several layers at once.

One starts off with three lives but can gain extra lives by obtaining a large score, however there is another snag - one also gains more little

apples and even two other enemies, the Green Butterfly and the Mask of Death. Killing these is correspondingly more difficult but scores even higher points.

The game makes good use of the high-resolution graphics of the Apple and includes some sound effects, which could have been better used.

The review copy was loaned by Pete & Pam Computers of Streatham from whom the game is available at £18.34. Apple Panic was used each day at the recent PCW Show, on the Croydon ComputerTown stand. It was continuously popular with the children, many of whom came back time and again for another go. It has also been assessed at ComputerTown in Croydon. The children all rate it as a good game and typical comments included: "...provides a lot of fun..."; "...a good game ... close to the arcade games..."; "...witty ... game of both skill and chance..."; "...A good game for all ages..."; "...Compulsive but without the destructive excitement of Invaders..."

These comments are from children themselves, but I thoroughly agree with them.

Thanks must go to Pete & Pam Computers for lending the game and also to the children who helped by playing it, especially to Kevin, Alan, Douglas and Johnathan who took the trouble to write down their comments for me.

type-right

Turn your Apple II or IIT 2020 into a professional Word-Processing system with full upper and lower case display and proper shift-key operation.

Plug-in fitting, with enable/disable switch, so no conflict with any existing hardware or software.

Complete with manual, fitting instructions and software on disk (including Applewriter conversion, etc).

£46.50 inclusive of VAT, Postage, etc.

● **MICROSOURCE**

1 Branch Rd,
Park Street,
St. Albans.

Tel: Park Street (0727) 72917

NOW THE SOFTCARD™ CAN TAKE YOU BEYOND THE BASICS.

You probably know about the SoftCard — our ingenious circuit card that converts an Apple II® into a Z-80® machine running CP/M®.

You may even know that with the SoftCard, you get Microsoft's powerful BASIC — extended to support Apple graphics and many other features.

Now, whenever you're ready to get beyond the BASICs, the SoftCard can take you into whole new realms. Starting with two advanced language packages from Microsoft.

FORTRAN AND COBOL TO GO.

Now you can run the world's most popular engineering/scientific language and the most popular business language on your Apple. Think what that means: you can choose from literally thousands of "off-the-shelf" applications programs, and have them working with little conversion. Or design your own programs, taking advantage of all the problem-solving power these specialized languages give you.

FORTRAN-80

A complete ANSI-standard FORTRAN (except COMPLEX type), with important enhancements. The extremely fast compiler performs extensive code

optimization, and since it doesn't require a "compiler" interpreter at run time, your programs will typically execute 2-3 times faster than with Apple FORTRAN.

FORTRAN is easy to learn if you know BASIC, and the package includes a huge library of floating point, math, and I/O routines you can use in all your programs.

COBOL-80

Virtually the only choice for serious business data processing.

It's ANSI 1974 standard COBOL, with many user-oriented features added: formatted screen support for CRT terminals, simple segmenting of very large programs, powerful file handling capability, trace debugging, and much more. A separate Sort package is coming soon.

FORTRAN-80 and COBOL-80 are just two more reasons why the Apple with SoftCard is the world's most versatile personal computer. Get all the exciting details from your Microsoft dealer today. And start getting beyond the BASICs.

Softcard and Microsoft are trademarks of Microsoft, Inc. Apple II is a registered trademark of Apple Computer, Inc. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc. Microsoft Consumer Products is a division of Microsoft, Inc.

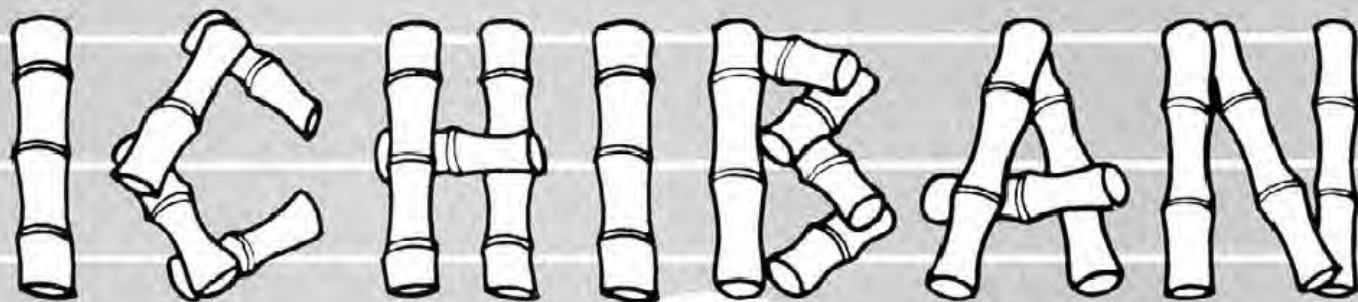


Pete & Pam Computers

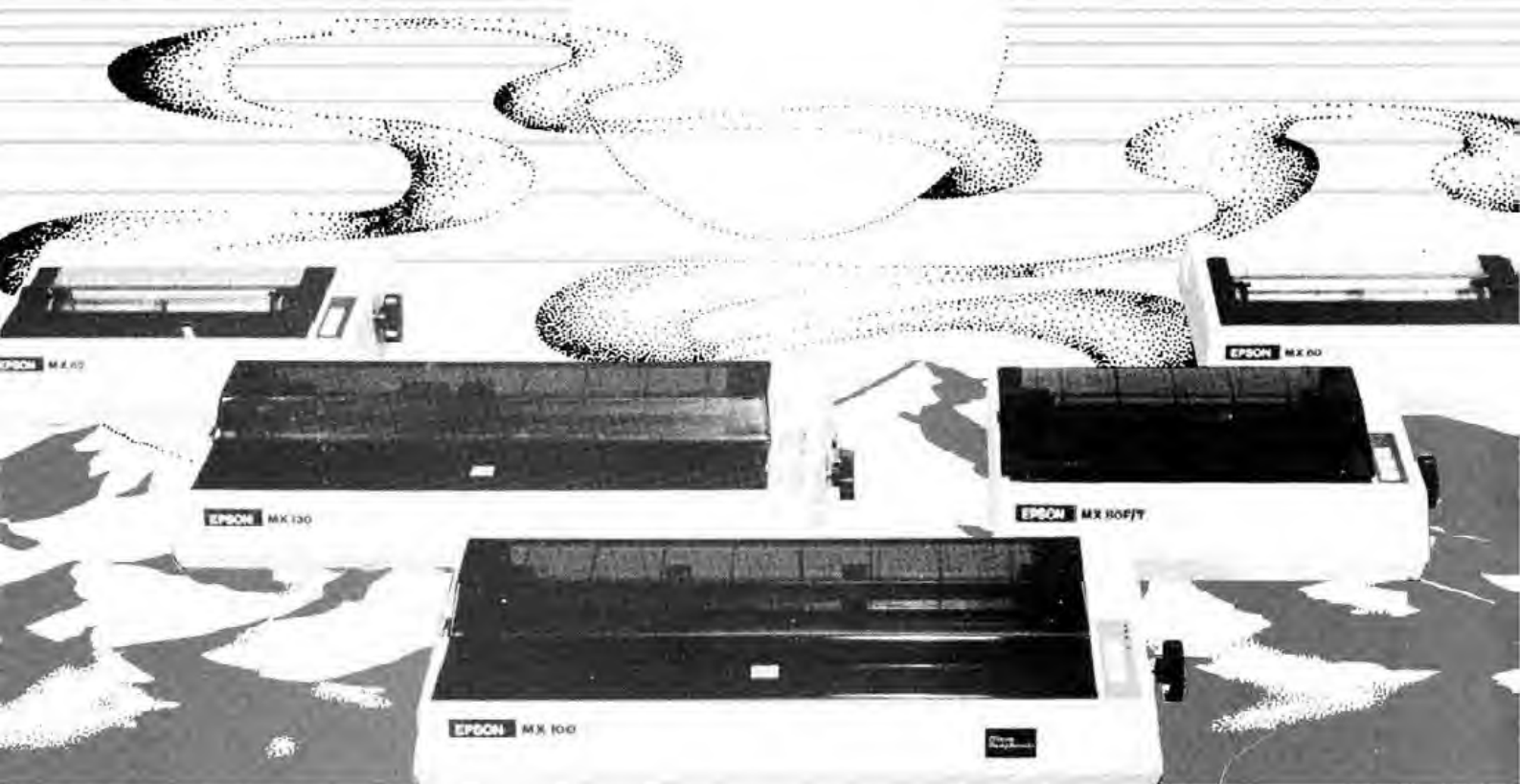
Waingate Lodge,
Waingate Close,
Rossendale, Lancs., BB4 7SQ
Telephone Rossendale (0706) 227011

MICROSOFT™
CONSUMER PRODUCTS

400 108th Ave. N.E., Suite 200
Bellevue, WA 98004 U.S.A.
(206) 454-1315



NUMBER ONE IN JAPANESE



Highest possible quality + lowest possible cost = EPSON + MICRO PERIPHERALS
for the best of both worlds.

	Matrix			Paper Handling		Speed		Max. Paper Width		High Res. Graphics & Italics
	7 x 5	9 x 7	9 x 9	Friction	Tractor	80 CPS	135 CPS	10"	15 1/2"	
MX80 T			●		●	●		●		
MX80 FT/1			●	●	●	●		●		
MX80 FT/New Type 2			●	●	●	●		●		●
MX82			●	●	●	●		●		●
MX100			●	●	●	●			●	●
MX130		●		●	●		●		●	●

The above machines have many more features including interfaces for Apple, PET (with PET Graphics), TRS80, Sharp, NEC, Hitachi, Nacom, Acorn, Super Brain, Video Genie, BBC Micro etc, some have correspondence quality printing and multiple character sets including international languages. Ring Ian today for full details and specifications and printout samples. All machines usually ex-stock with next day delivery plus 12 months no-quibble guarantee.

Wholesalers and Distributors of quality Japanese Micro Products.

Micro Peripherals Ltd.

61 NEW MARKET SQUARE, BASINGSTOKE, HANTS. Telephone: 0256 56468

Europe's leading Epson Distributor and Supplier. Trade Sales - 0256 56469

JAPANESE OFFICE: 101 Abe Bldg. 4F, 2-42 Kanda Jinbocho, Chiyado-ku, Tokyo, Japan.